NS9215

Hardware Reference

90000847_C

Release date: 10 April 2008

Downloaded from www.Manualslib.com manuals search engine

©2008 Digi International Inc.
Printed in the United States of America. All rights reserved.

Digi, Digi International, the Digi logo, a Digi International Company, ConnectCore, NET+, NET+OS and NET+Works are trademarks or registered trademarks of Digi International, Inc. in the United States and other countries worldwide. All other trademarks are the property of their respective owners.

Information in this document is subject to change without notice and does not represent a commitment on the part of Digi International.

Digi provides this document "as is," without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose. Digi may make improvements and/or changes in this manual or in the product(s) and/or the program(s) described in this manual at any time.

This product could include technical inaccuracies or typographical errors. Changes are made periodically to the information herein; these changes may be incorporated in new editions of the publication.

Downloaded from www.Manualslib.com manuals search engine

Downloaded from www.Manualslib.com manuals search engine

Downloaded from www.Manualslib.com manuals search engine

Contents

Chapter 1: Pinout (265) 27
The Legend 27
Memory bus interface 28
Ethernet interface MAC 30
General purpose I/O (GPIO) 31
System clock 43
System clock drawing 44
RTC clock and battery backup drawing 45
System mode 45
System reset 47
JTAG Test 48
ADC 49
POR and battery-backed logic 50
Power and ground 51

Chapter 2: I/O Control Module 53
System memory bus I/O control 53
Control and Status registers 53
Register address map 53
GPIO Configuration registers 55
GPIO configuration options 55
GPIO Configuration Register #0 56
GPIO Configuration Register #1 56
GPIO Configuration Register #2 57
GPIO Configuration Register #3 57
GPIO Configuration Register #4 58
GPIO Configuration Register #5 58
GPIO Configuration Register #6 59
GPIO Configuration Register #7 59
GPIO Configuration Register #8 60
GPIO Configuration Register #9 60
GPIO Configuration Register #10 61
GPIO Configuration Register #11 61
GPIO Configuration Register #12 62
GPIO Configuration Register #13 62
GPIO Configuration Register #14 63

Downloaded from www.Manualslib.com manuals search engine

GPIO Configuration Register #15 63
GPIO Configuration Register #16 64
GPIO Configuration Register #17 64
GPIO Configuration Register #18 65
GPIO Configuration Register #19 65
GPIO Configuration Register #20 66
GPIO Configuration Register #21 66
GPIO Configuration Register #22 67
GPIO Configuration Register #23 67
GPIO Configuration Register #24 68
GPIO Configuration Register #25 68
GPIO Configuration Register #26 69
GPIO Control registers 70
GPIO Control Register #0 70
GPIO Control Register #1 71
GPIO Control Register #2 72
GPIO Control Register #3 73
GPIO Status registers 74
GPIO Status Register #1 74
GPIO Status Register #2 75
GPIO Status Register #3 76
Memory Bus Configuration register 76

Chapter 3: Working with the CPU 81
About the processor 81
Arm926EJ-S process block diagram 82
Instruction sets 82
ARM instruction set 82
Thumb instruction set 82
Java instruction set 83
System control processor (CP15) registers 83
ARM926EJ-S system addresses 83
Address manipulation example 83
Accessing CP15 registers 83
Terms and abbreviations 84
Register summary 85
R0: ID code and cache type status registers 86
R0: ID code 86
R0: Cache type register 86
Cache type register and field description 87
Dsize and Isize fields 87
R1: Control register 88
Control register 89
Bit functionality 89

Downloaded from www.Manualslib.com manuals search engine

ICache and DCache behavior 90
R2: Translation Table Base register 91
Register format 91
R3:Domain Access Control register 91
Register format 91
Access permissions and instructions 91
R4 register 92
R5: Fault Status registers 92
Access instructions 92
Register format 92
Register bits 92
Status and domain fields 93
R6: Fault Address register 93
Access instructions 93
R7:Cache Operations register 94
Write instruction 94
Cache functions 94
Cache operation functions 95
Modified virtual address format (MVA) 96
Set/Way format 96
Set/Way example 96
Test and clean DCache instructions 96
Test, clean, and invalidate DCache instruction 97
R8:TLB Operations register 97
TLB operations 97
TLB operation instructions 97
Modified virtual address format (MVA) 98
R9: Cache Lockdown register 98
Cache ways 98
Instruction or data lockdown register 99
Access instructions 99
Modifying the Cache Lockdown register 99
Register format 99
Cache Lockdown register L bits 99
Lockdown cache: Specific loading of addresses into a cache-way 100
Cache unlock procedure 101
R10:TLB Lockdown register 101
Register format 101
P bit 101
Invalidate operation 101
Programming instructions 102
Sample code sequence 102
R11 and R12 registers 102
R13:Process ID register 102
FCSE PID register 103

Downloaded from www.Manualslib.com manuals search engine

Access instructions 103
Register format 103
Performing a fast context switch 103
Context ID register 104
Access instructions 104
Register format 104
R14 register 104
R15: Test and debug register 104
Jazelle(Java) 104
DSP 105
MemoryManagement Unit (MMU) 105
MMU Features 105
Access permissions and domains 106
Translated entries 106
MMU program accessible registers 107
Address translation 107
Translation table base 108
TTB register format 108
Table walk process 109
First-level fetch 109
First-level fetch concatenation and address 110
First-level descriptor 110
Page table descriptors 110
First-level descriptor bit assignments: Priority encoding of fault status 111
First-level descriptor bit assignments: Interpreting first level descriptor bits [1:0] 111
Section descriptor 111
Section descriptor format 111
Section descriptor bit description 112
Coarse page table descriptor 112
Coarse page table descriptor format 112
Coarse page table descriptor bit description 112
Fine page table descriptor 112
Fine page table descriptor format 113
Fine page table descriptor bit description 113
Translating section references 113
Second-level descriptor 114
Second-level descriptor format 114
Second-level descriptor pages 114
Second-level descriptor bit assignments 115
Second-level descriptor least significant bits 115
Translation sequence for large page references 116
Translating sequence for small page references 117
Translation sequence for tiny page references 118
Subpages 118

Downloaded from www.Manualslib.com manuals search engine

MMU faults and CPU aborts 119
Alignment fault checking 119
Fault Address and Fault Status registers 119
Priority encoding table 120
Fault Address register (FAR) 120
FAR values for multi-word transfers 120
Compatibility issues 121
Domain access control 121
Specifying access permissions 121
Interpreting access permission bits 121
Fault checking sequence 122
Alignment faults 123
Translation faults 124
Domain faults 124
Permission faults 124
External aborts 125
Enabling and disabling the MMU 125
Enabling the MMU 125
Disabling the MMU 126
TLB structure 126
Caches and write buffer 127
Cache features 127
Write buffer 128
Enabling the caches 128
ICache I and M bit settings 129
ICache page table C bit settings 129
R1 register C and M bits for DCache 129
DCache page table C and B settings 129
Cache MVA and Set/Way formats 130
Generic, virtually indexed, virtually addressed cache 131
ARM926EJ-S cache format 132
ARM926EJ-S cache associativity 132
Set/way/word format for ARM926EJ-S caches 132
Noncachable instruction fetches 133
Self-modifying code 133
AHB behavior 134
Instruction Memory Barrier 134
IMB operation 134
Sample IMB sequences 135

Chapter 4: System Control Module 137
Features 137
Bus interconnection 137
System bus arbiter 138

Downloaded from www.Manualslib.com manuals search engine

High speed bus system 138
High-speed bus arbiters 138
How the bus arbiter works 138
Ownership 139
Locked bus sequence 139
Relinquishing the bus 139
SPLIT transfers 140
Arbiter configuration example 140
Address decoding 141
Programmable timers 142
Software watchdog timer 142
General purpose timers/counters 143
Source clock frequency 143
GPTC characteristics 143
Control field 143
16-bit mode options 144
Basic PWM function 144
Functional block diagram 144
Enhanced PWM function 145
Sample enhanced PWM waveform 145
Quadrature decoder function 145
How the quadrature decoder/counter works 146
Provides input signals 146
Monitors how far the encoder has moved 147
Digital filter 147
Testing signals 147
Timer support 147
Interrupt controller 148
FIQ interrupts 148
IRQ interrupts 148
32-vector interrupt controller 148
IRQ characteristics 149
Interrupt sources 149
Vectored interrupt controller (VIC) flow 151
Configurable system attributes 151
PLL configuration 151
PLL configuration and control system block diagram 152
Bootstrap initialization 152
Configuring the powerup settings 152
System configuration registers 154
Register address map 154
General Arbiter Control register 158
BRC0, BRC1, BRC2, and BRC3 registers 158
Channel allocation 159
AHB Error Detect Status 1 159

Downloaded from www.Manualslib.com manuals search engine

AHB Error Detect Status 2 160
AHB Error Monitoring Configuration register 161
Timer Master Control register 162
Timer 0–4 Control registers..... 164
Timer 5 Control register 166
Timer 6–9 Control registers..... 168
Timer 6–9 High registers 170
Timer 6–9 Low registers..... 171
Timer 6–9 High and Low Step registers 172
Timer 6–9 Reload Step registers 172
Timer 0-9 Reload Count and Compare register 173
Timer 0-9 Read and Capture register 174
Interrupt Vector Address Register Level 31–0 175
Int (Interrupt) Config (Configuration) 31–0 registers 175
Individual register mapping 175
ISADDR register..... 176
Interrupt Status Active..... 177
Interrupt Status Raw 178
Software Watchdog Configuration 178
Software Watchdog Timer 179
Clock Configuration register 180
Module Reset register 182
Miscellaneous System Configuration and Status register 184
PLL Configuration register..... 186
PLL frequency formula 186
Active Interrupt Level ID Status register 187
Power Management..... 187
AHB Bus Activity Status 190
System Memory Chip Select 0 Dynamic Memory Base and Mask registers..... 190
System Memory Chip Select 1 Dynamic Memory Base and Mask registers..... 191
System Memory Chip Select 2 Dynamic Memory Base and Mask registers..... 192
System Memory Chip Select 3 Dynamic Memory Base and Mask registers..... 193
System Memory Chip Select 0 Static Memory Base and Mask registers 194
System Memory Chip Select 1 Static Memory Base and Mask registers 195
System Memory Chip Select 2 Static Memory Base and Mask registers 196
System Memory Chip Select 3 Static Memory Base and Mask registers 197
Gen ID register 198
External Interrupt 0–3 Control register 199
RTC Module Control register 200

Chapter 5: Memory Controller 203
Features 203
Low-power operation..... 204
Low-power SDRAM deep-sleep mode..... 204

Downloaded from www.Manualslib.com manuals search engine

Low-power SDRAM partial array refresh 204
Memory map....205
Power-on reset memory map 205
Chip select 1 memory configuration....205
Example: Boot from flash, SRAM mapped after boot 205
Example: Boot from flash, SDRAM remapped after boot 206
Static memory controller....207
Write protection 208
Extended wait transfers 208
Memory mapped peripherals....209
Static memory initialization 209
Access sequencing and memory width 209
Wait state generation....209
Programmable enable....210
Static memory read control....210
Output enable programmable delay....210
ROM, SRAM, and Flash 210
Static memory read: Timing and parameters 211
External memory read transfer with zero wait states 211
External memory read transfer with two wait states....211
External memory read transfer with two output enable delay states....212
External memory read transfers with zero wait states 212
Burst of zero wait states with fixed length....213
Burst of two wait states with fixed length 213
Asynchronous page mode read 214
Asynchronous page mode read: Timing and parameters 214
External memory page mode read transfer 214
External memory 32-bit burst read from 8-bit memory 215
Static memory write control....216
Write enable programming delay 216
SRAM....216
Static memory Write: Timing and parameters....216
External memory write transfer with zero wait states 216
External memory write transfer with two wait states....217
External memory write transfer with two write enable delay states 217
Two external memory write transfers with zero wait states 218
Flash memory 218
Bus turnaround....219
Bus turnaround: Timing and parameters....219
Read followed by write with no turnaround....219
Write followed by a read with no turnaround....220
Read followed by a write with two turnaround cycles....220
Byte lane control 221
Address connectivity 222
Memory banks constructed from 8-bit or non-byte-partitioned memory devices

Downloaded from www.Manualslib.com manuals search engine

222
Memory banks constructed from 16-or 32-bit memory devices 223
Dynamic memory controller 225
Write protection 225
Access sequencing and memory width 225
SDRAM Initialization 225
Left-shift value table: 32-bit wide data bus SDRAM (RBC) 226
Left-shift value table: 32-bit wide data bus SDRAM (BRC) 227
Left-shift value table: 16-bit wide data bus SDRAM (RBC) 227
Left-shift value table: 16-bit wide data bus SDRAM (BRC) 228
SDRAM address and data bus interconnect 228
32-bit wide configuration 228
32-bit wide configuration 229
Registers 230
Register map 230
Reset values 232
Control register 232
Status register 234
Configuration register 234
Dynamic Memory Control register 235
Dynamic Memory Refresh Timer register 236
Register 237
Dynamic Memory Read Configuration register 237
Dynamic Memory Precharge Command Period register 238
Dynamic Memory Active to Precharge Command Period register 239
Dynamic Memory Self-refresh Exit Time register 240
Dynamic Memory Last Data Out to Active Time register 240
Dynamic Memory Data-in to Active Command Time register 241
Dynamic Memory Write Recovery Time register 242
Dynamic Memory Active to Active Command Period register 243
Dynamic Memory Auto Refresh Period register 243
Dynamic Memory Exit Self-refresh register 244
Dynamic Memory Active Bank A to Active Bank B Time register 245
Dynamic Memory Load Mode register to Active Command Time register 246
Static Memory Extended Wait register 247
Example 247
Dynamic Memory Configuration 0–3 registers 247
Address mapping for the Dynamic Memory Configuration registers 249
Chip select and memory devices 250
Dynamic Memory RAS and CAS Delay 0–3 registers 250
StaticMemory Configuration 0–3 registers 251
StaticMemory Write Enable Delay 0–3 registers 254
Static Memory Output Enable Delay 0–3 registers 255
Static Memory Read Delay 0–3 registers 256
StaticMemory Page Mode Read Delay 0–3 registers 256

Downloaded from www.Manualslib.com manuals search engine

Static Memory Write Delay 0–3 registers....257
StaticMemory Turn Round Delay 0–3 registers....258

Chapter 6: Ethernet Communication Module 261
Features....261
Common acronyms....261
Ethernet communications module....262
Ethernet MAC....262
MAC module block diagram....263
MAC module features....263
PHY interface mappings....264
Station address logic (SAL)....264
MAC receiver....265
Statistics module....265
Ethernet front-end module....266
Ethernet front-end module (EFE)....266
Receive packet processor....266
Transmit packet processor....267
Receive packet processor....267
Power down mode....267
Transferring a frame to system memory....268
Receive buffer descriptor format....268
Receive buffer descriptor format description....268
Receive buffer descriptor field definitions....269
Transmit packet processor....269
Transmit buffer descriptor format....270
Transmit buffer descriptor field definitions....270
Transmitting a frame....271
Frame transmitted successfully....272
Frame transmitted unsuccessfully....272
Transmitting a frame to the Ethernet MAC....272
Ethernet underrun....272
Ethernet slave interface....273
Interrupts....273
Interrupt sources....273
Status bits....274
Resets....274
Multicast address filtering....275
Filter entries....275
Multicast address filter registers....275
Multicast address filtering example 1....275
Multicast address filtering example 2....276
Notes....276
Clock synchronization....276

Downloaded from www.Manualslib.com manuals search engine

Writing to other registers 276
Ethernet Control and Status registers 277
Register address filter 277
Ethernet General Control Register #1 279
Ethernet General Control Register #2 282
Ethernet General Status register 283
Ethernet Transmit Status register 284
Ethernet Receive Status register 286
MAC Configuration Register #1 288
MAC Configuration Register #2 289
PAD operation table for transmit frames 291
Back-to-Back Inter-Packet-Gap register 291
Non Back-to-Back Inter-Packet-Gap register 292
Collision Window/Retry register 293
Maximum Frame register 294
MII Management Configuration register 295
Clocks field settings 296
MII Management Command register 296
MII Management Address register 297
MII Management Write Data register 298
MII Management Read Data register 298
MII Management Indicators register 299
Station Address registers 300
Station Address Filter register 301
RegisterHash Tables 302
HT1 302
HT2 303
Statistics registers 303
Combined transmit and receive statistics counters address map 303
Receive statistics counters address map 304
Receive byte counter (A060 069C) 304
Receive packet counter (A060 06A0) 304
Receive FCS error counter (A060 06A4) 305
Receive multicast packet counter (A060 06A8) 305
Receive broadcast packet counter (A060 06AC) 305
Receive control frame packet counter (A060 06B0) 305
Receive PAUSE frame packet counter (A060 06B4) 305
Receive unknown OPCODE packet counter (A060 06B8) 305
Receive alignment error counter (A060 06BC) 306
Receive code error counter (A060 06C4) 306
Receive carrier sense error counter (A060 06C8) 306
Receive undersize packet counter (A060 06CC) 306
Receive oversize packet counter (A060 06D0) 306
Receive fragments counter (A060 06D4) 306
Receive jabber counter (A060 06D8) 307

Downloaded from www.Manualslib.com manuals search engine

Transmit statistics counters address map 307
Transmit byte counter (A060 06E0) 307
Transmit packet counter (A060 06E4) 308
Transmit multicast packet counter (A060 06E8) 308
Transmit broadcast packet counter (A060 06EC) 308
Transmit deferral packet counter (A060 06F4) 308
Transmit excessive deferral packet counter (A060 06F8) 308
Transmit single collision packet counter (A060 06FC) 308
Transmit multiple collision packet counter (A060 0700) 309
Transmit late collision packet counter (A060 0704) 309
Transmit excessive collision packet counter (A060 0708) 309
Transmit total collision packet counter (A060 070C) 309
Transmit jabber frame counter (A060 0718) 309
Transmit FCS error counter (A060 071C) 309
Transmit oversize frame counter (A060 0724) 310
Transmit undersize frame counter (A060 0728) 310
Transmit fragment counter (A060 072C) 310
General Statistics registers address map 310
Carry Register 1 310
Carry Register 2 311
Carry Register 1 Mask register 312
Carry Register 2 Mask register 314
RX_A Buffer Descriptor Pointer register 315
RX_B Buffer Descriptor Pointer register 315
RX_C Buffer Descriptor Pointer register 316
RX_D Buffer Descriptor Pointer register 316
Ethernet Interrupt Status register 317
Ethernet Interrupt Enable register 319
TX Buffer Descriptor Pointer register 320
Transmit Recover Buffer Descriptor Pointer register 321
TX Error Buffer Descriptor Pointer register 321
TX Stall Buffer Descriptor Pointer register 322
RX_A Buffer Descriptor Pointer Offset register 323
RX_B Buffer Descriptor Pointer Offset register 324
RX_C Buffer Descriptor Pointer Offset register 324
RX_D Buffer Descriptor Pointer Offset register 325
Transmit Buffer Descriptor Pointer Offset register 325
RX Free Buffer register 326
Multicast Address Filter registers 327
Multicast Low Address Filter Register #0 327
Multicast Low Address Filter Register #1 327
Multicast Low Address Filter Register #2 327
Multicast Low Address Filter Register #3 327
Multicast Low Address Filter Register #4 327
Multicast Low Address Filter Register #5 327

Downloaded from www.Manualslib.com manuals search engine

Multicast Low Address Filter Register #6 328
Multicast Low Address Filter Register #7 328
Multicast High Address Filter Register #0 328
Multicast High Address Filter Register #1 328
Multicast High Address Filter Register #2 328
Multicast High Address Filter Register #3 328
Multicast High Address Filter Register #4 328
Multicast High Address Filter Register #5 328
Multicast High Address Filter Register #6 329
Multicast High Address Filter Register #7 329
Multicast Address Mask registers 329
Multicast Low Address Mask Register #0 329
Multicast Low Address Mask Register #1 329
Multicast Low Address Mask Register #2 329
Multicast Low Address Mask Register #3 329
Multicast Low Address Mask Register #4 330
Multicast Low Address Mask Register #5 330
Multicast Low Address Mask Register #6 330
Multicast Low Address Mask Register #7 330
Multicast High Address Mask Register #0 330
Multicast High Address Mask Register #1 330
Multicast High Address Mask Register #2 330
Multicast High Address Mask Register #3 330
Multicast High Address Mask Register #4 330
Multicast High Address Mask Register #5 331
Multicast High Address Mask Register #6 331
Multicast High Address Mask Register #7 331
Multicast Address Filter Enable register 331
TX Buffer Descriptor RAM 332
Offset+0 332
Offset+4 333
Offset+8 333
Offset+C 333
RX FIFO RAM 333
Sample hash table code 334

Chapter 7: External DMA 339
DMA transfers 339
Initiating DMA transfers 339
Processor-initiated 339
External peripheral-initiated 339
DMA buffer descriptor 340
DMA buffer descriptor diagram 340
Source address [pointer] 340

Downloaded from www.Manualslib.com manuals search engine

Buffer length 340
Destination address [pointer] 340
Status 341
Wrap (W) bit 341
Interrupt (I) bit 341
Last (L) bit 341
Full (F) bit 341
Descriptor list processing 341
Peripheral DMA read access 342
Determining the width of PDEN 342
Equation variables 342
Peripheral DMA single read access 343
Peripheral DMA burst read access 343
Peripheral DMA write access 343
Determining the width of PDEN 344
Peripheral DMA single write access 344
Peripheral DMA burst write access 344
Peripheral REQ and DONE signaling 344
REQ signal 344
DONE signal 345
Special circumstances 345
Static RAM chip select configuration 345
Static ram chip select configuration 345
Control and Status registers 346
Register address map 346
DMA Buffer Descriptor Pointer 346
DMA Control register 347
DMA Status and Interrupt Enable register 350
DMA Peripheral Chip Select register 352

Chapter 8: AES Data Encryption/Decryption Module 355
Features 355
Block diagram 356
Data blocks 356
AES DMA buffer descriptor 356
AES buffer descriptor diagram 357
Source address [pointer] 357
Source buffer length 357
Destination buffer length 357
Destination address [pointer] 357
AES control 357
AES op code 358
WRAP (W) bit 358
Interrupt (I) bit 358

Downloaded from www.Manualslib.com manuals search engine

Last (L) bit 358
Full (F) bit 358
Decryption 359
ECB processing 359
Processing flow diagram 359
CBC, CFB, OFB, and CTR processing 360
Processing flow diagram 360
CCM mode 360
Nonce buffer 361
Processing flow 361

Chapter 9: I/O Hub Module 363
Block diagram 364
AHB slave interface 364
DMA controller 364
Servicing RX and FIFOs 364
Buffer descriptors 365
Source address [pointer] 365
Buffer length 365
Control[15] – W 365
Control[14] – I 365
Control[13] – L 365
Control[12] – F 365
Control[11:0] 366
Status[15:0] 366
Transmit DMA example 367
Process 367
Visual example 368
Control and status register address maps 368
UART A register address map 369
UART B register address map 369
UART C register address map 370
UART D register address map 370
SPI register address map 371
AD register address map 371
Reserved 371
I2C register address map 371
Reserved 371
RTC register address map 372
IO Hardware Assist register address map (0) 372
IO Hardware Assist register address map (1) 372
IO register address map (0) 372
IO register address map (1) 372
[Module] Interrupt and FIFO Status register 372

Downloaded from www.Manualslib.com manuals search engine

[Module] DMA RX Control 375
[Module] DMA RX Buffer Descriptor Pointer 376
[Module] RX Interrupt Configuration register 377
[Module] Direct Mode RX Status FIFO 378
[Module] Direct Mode RX Data FIFO 379
[Module] DMA TX Control 380
[Module] DMA TX Buffer Descriptor Pointer 381
[Module] TX Interrupt Configuration register 381
[Module] Direct Mode TX Data FIFO 382
[Module] Direct Mode TX Data Last FIFO 383

Chapter 10: Serial Control Module: UART 385
Features 385
UART module structure 386
Normal mode operation 386
Example configuration 386
Baud rate generator 387
Baud rates 387
Hardware-based flow control 388
Character-based flow control (XON/XOFF) 388
Example configuration 388
Forced character transmission 388
Force character transmission procedure 389
Collecting feedback 389
ARM wakeup on character recognition 389
Example configuration 389
Wrapper Control and Status registers 390
Register address map 390
Wrapper Configuration register 391
Interrupt Enable register 393
Interrupt Status register 395
Receive Character GAP Control register 398
Receive Buffer GAP Control register 399
Receive Character Match Control register 399
Receive Character-Based Flow Control register 400
Force Transmit Character Control register 402
ARM Wakeup Control register 403
Transmit Byte Count 404
UART Receive Buffer 405
UART Transmit Buffer 405
UART Baud Rate Divisor LSB 406
UART Baud Rate Divisor MSB 406
UART Interrupt Enable register 407
UART Interrupt Identification register 408

Downloaded from www.Manualslib.com manuals search engine

UART FIFO Control register 409
UART Line Control register 409
UART Modem Control register 411
UART Line Status register 411
UART Modem Status register 412

Chapter 11: Serial Control Module: HDLC 415
HDLC module structure 415
Receive and transmit operations 415
Receive operation 416
Transmit operation 416
Transmitter underflow 416
Clocking 416
Bits 416
Last byte bit pattern table 417
Data encoding 417
Encoding examples 417
Digital phase-locked-loop (DPLL) operation: Encoding 418
Transitions 418
DPLL-tracked bit cell boundaries 419
NRZ and NRZI data encoding 419
Biphase data encoding 419
DPLL operation: Adjustment ranges and output clocks 419
NRZ and NRZI encoding 420
Biphase-Level encoding 420
Biphase-Mark and Biphase-Space encoding 421
IRDA-compliant encode 421
Normal mode operation 421
Example configuration 421
Wrapper and HDLC Control and Status registers 422
Register address map 422
Wrapper Configuration register 422
Interrupt Enable register 424
Interrupt Status register 425
HDLC Data Register 1 427
HDLC Data Register 2 427
HDLC Data register 3 428
HDLC Control Register 1 429
HDLC Control Register 2 429
HDLC Clock Divider Low 430
HDLC Clock Divider High 431

Chapter 12: Serial Control Module: SPI 433
Features 433

Downloaded from www.Manualslib.com manuals search engine

SPI module structure 434
SPI controller 434
Simple parallel/serial data conversion 434
Full duplex operation 434
SPI clocking modes 435
Timing modes 435
Clocking mode diagrams 435
SPI clock generation 436
Clock generation samples 436
In SPI master mode 436
In SPI slave mode 436
System boot-over-SPI operation 436
Available strapping options 437
EEPROM/FLASH header 437
Header format 437
Time to completion 438
SPI Control and Status registers 439
Register address map 439
SPI Configuration register 439
Clock Generation register 440
Register programming steps 441
Interrupt Enable register 441
Interrupt Status register 442
SPI timing characteristics 443
SPI master timing diagram 444
SPI slave timing parameters 444
SPI slave timing diagram 445

Chapter 13: I2C Master/Slave Interface 447
Overview 447
Physical I2C bus 447
Multi-master bus 448
I2C external addresses 448
I2C command interface 449
Locked interrupt driven mode 449
Master module and slave module commands 449
Bus arbitration 449
I2C registers 450
Register address map 450
Command Transmit Data register 450
Register 450
Register bit assignment 451
Status Receive Data register 451
Register 451

Downloaded from www.Manualslib.com manuals search engine

Register bit assignment 451
Master Address register 452
Register 452
Register bit assignment 453
Slave Address register 453
Register 453
Register bit assignment 453
Configuration register 454
Timing parameter for fast-mode 454
Register 454
Register bit assignment 454
Interrupt Codes 455
Master/slave interrupt codes 455
Software driver 456
I2C master software driver 456
I2C slave high level driver 456
Flow charts 457
Master module (normal mode, 16-bit) 457
Slave module (normal mode, 16-bit) 458

Chapter 14: Real Time Clock Module 459
RTC functionality 459
RTC configuration and status registers 460
Register address map 460
RTC General Control register 460
12/24 Hour register 461
Time register 462
Calendar register 463
Time Alarm register 464
Calendar Alarm register 465
Alarm Enable register 465
Event Flags register 466
Interrupt Enable register 468
Interrupt Disable register 469
Interrupt Enable Status register 470
General Status register 471

Chapter 15: Analog-to-Digital Converter (ADC) Module 473
Features 473
ADC module structure 473
ADC control block 474
ADC DMA procedure 474
ADC control and status registers 475
Register address map 475

Downloaded from www.Manualslib.com manuals search engine

ADC Configuration register 475
ADC Clock Configuration register 477
ADC Output Registers 0-7 477

Chapter 16: Timing 479
Electrical characteristics 479
Absolute maximum ratings 479
Recommended operating conditions 480
Power dissipation 480
DC electrical characteristics 481
Inputs 481
Ouputs 482
Reset and edge sensitive input timing requirements 482
..... 483
Memory Timing 484
SDRAM burst read (16-bit) 485
SDRAM burst read (16 bit), CAS latency = 3 486
SDRAM burst write (16 bit) 487
SDRAM burst read (32 bit) 488
SDRAM burst read (32 bit), CAS latency = 3 489
SDRAM burst write (32-bit) 490
SDRAM load mode 491
SDRAM refresh mode 492
Clock enable timing 493
Values in SRAM timing diagrams 494
Static RAM read cycles with 0 wait states 495
Static RAM asynchronous page mode read, WTPG = 1 496
Static RAM read cycle with configurable wait states 497
Static RAM sequential write cycles 498
Static RAM write cycle 499
Static write cycle with configurable wait states 500
Slow peripheral acknowledge timing 501
Slow peripheral acknowledge read 502
Slow peripheral acknowledge write 502
Ethernet timing 503
Ethernet MII timing 503
I^2C timing 504
SPI Timing 505
SPI master mode 0 and 1: 2-byte transfer 507
SPI master mode2 and 3: 2-byte transfer 507
SPI slave mode 0 and 1: 2-byte transfer 508
SPI slave mode 2 and 3: 2-byte transfer 508
Reset and hardware strapping timing 509
JTAG timing 510

Downloaded from www.Manualslib.com manuals search engine

Clock timing 511
System PLL reference clock timing..... 511

Chapter 17: Packaging 513
Package..... 513
Processor Dimensions 514

Chapter 18: Change log 517
Revision B 517
Revision C 517

Downloaded from www.Manualslib.com manuals search engine

Downloaded from www.Manualslib.com manuals search engine

Pinout (265)

CHAPTER 1

The NS9215 offers a connection to a 10/100 Ethernet network, as well as a glueless connection to SDRAM, PC100 DIMM, flash, EEPROM, and SRAM memories, and an external bus expansion module. It includes four multi-function serial ports, one I2C channel, 12-bit Analog to Digital converter, battery backed real time clock and an AES data encryption/decryption module. The NS215 provides up to 108 general purpose I/O (GPIO) pins and configurable power management with sleep mode.

The Legend

Heading	Description
Pin	Pin number assigned for a specific I/O signal
Signal	Pin name for each I/O signal. Some signals have multiple function modes and are identified accordingly. The mode is configured through firmware using one or more configuration registers. _n is the signal name indicates that this signal is active is active *low*.
U/D	U or D indicates whether the pin has an internal pullup resistor or a pulldown resistor: U Pullup (input current source) D Pulldown (input current sink) If no value is listed, that pin has neither an internal pullup nor pulldown resistor.
I/O	The type of signal: input (I), output (O), input/output (I/O), or power (P).
OD (mA)	The output drive of an output buffer. The NS9215 uses one of two drivers: ■ 2 mA ■ 4 mA

Downloaded from www.Manualslib.com manuals search engine

Memory bus interface

Pin	Signal	U/D	I/O	OD	Description
B9	clk_out[0]		O	4	SDRAM bus clock
A15	clk_out[1]		O	4	SDRAM bus clock
P12	addr[27] / gpio_a[3][a]	U	I/O	4	Address bus, Endian
T14	addr[26] / gpio_a[2][a.]	U	I/O	4	Address bus, SPI boot
U15	addr[25] / gpio_a[1][a.]	U	I/O	4	Address bus
R12	addr[24] / gpio_a[0][a.]	U	I/O	4	Address bus, Boot width [1]
T13	addr[23]	U	I/O	4	Address bus, Boot width [0]
U14	addr[22]		O	4	Address bus
T12	addr[21]		O	4	Address bus
U13	addr[20]		O	4	Address bus,
R11	addr[19]	U	I/O	4	Address bus, GENID 10
T11	addr[18]	U	I/O	4	Address bus, GENID 9
U12	addr[17]	U	I/O	4	Address bus, GENID 8
T10	addr[16]	U	I/O	4	Address bus, GENID 7
R9	addr[15]	U	I/O	4	Address bus, GENID 6
U11	addr[14]	U	I/O	4	Address bus, GENID 5
U10	addr[13]	U	I/O	4	Address bus, GENID 4
T9	addr[12]	U	I/O	4	Address bus, GENID 3
U9	addr[11]	U	I/O	4	Address bus, GENID 2
U8	addr[10]	U	I/O	4	Address bus, GENID 1
T8	addr[9]	U	I/O	4	Address bus, GENID 0
U7	addr[8]	U	I/O	4	Address bus
T7	addr[7]	U	I/O	4	Address bus, PLL bypass
U6	addr[6]	U	I/O	4	Address bus, PLL OD [1]
T6	addr[5]	U	I/O	4	Address bus, PLL OD [0]
U5	addr[4]	U	I/O	4	Address bus, PLL NR[4]
M2	addr[3]	U	I/O	4	Address bus, PLL NR[3]
N1	addr[2]	U	I/O	4	Address bus, PLL NR[2]
L2	addr[1]	U	I/O	4	Address bus, PLL NR[1]

Downloaded from www.Manualslib.com manuals search engine

Pin	Signal	U/D	I/O	OD	Description
M1	addr[0]	U	I/O	4	Address bus, PLL NR[0]
L1	data[31]	U	I/O	4	Data bus
K2	data[30]	U	I/O	4	Data bus
K1	data[29]	U	I/O	4	Data bus
J1	data[28]	U	I/O	4	Data bus
J2	data[27]	U	I/O	4	Data bus
H1	data[26]	U	I/O	4	Data bus
G1	data[25]	U	I/O	4	Data bus
J3	data[24]	U	I/O	4	Data bus
H2	data[23]	U	I/O	4	Data bus
F1	data[22]	U	I/O	4	Data bus
G2	data[21]	U	I/O	4	Data bus
H3	data[20]	U	I/O	4	Data bus
E1	data[19]	U	I/O	4	Data bus
F2	data[18]	U	I/O	4	Data bus
D1	data[17]	U	I/O	4	Data bus
E2	data[16]	U	I/O	4	Data bus
H4	data[15] / gpio[31][b]	U	I/O	4	Data bus
G3	data[14] / gpio[30]	U	I/O	4	Data bus
G4	data[13] / gpio[29]	U	I/O	4	Data bus
G5	data[12] / gpio[28]	U	I/O	4	Data bus
F3	data[11] / gpio[27]	U	I/O	4	Data bus
F4	data[10] / gpio[26]	U	I/O	4	Data bus
F5	data[9] / gpio[25]	U	I/O	4	Data bus
C1	data[8] / gpio[24]	U	I/O	4	Data bus
E4	data[7] / gpio[23]	U	I/O	4	Data bus
D2	data[6] / gpio[22]	U	I/O	4	Data bus
E3	data[5] / gpio[21]	U	I/O	4	Data bus
B1	data[4] / gpio[20]	U	I/O	4	Data bus
D4	data[3] / gpio[19]	U	I/O	4	Data bus
C2	data[2] / gpio[18]	U	I/O	4	Data bus
B2	data[1] / gpio[17]	U	I/O	4	Data bus
D3	data[0] / gpio[16]	U	I/O	4	Data bus

Downloaded from www.Manualslib.com manuals search engine

Pin	Signal	U/D	I/O	OD	Description
A10	data_mask[3]		O	4	byte_enable data[31:24}
B11	data_mask[2]		O	4	Byte enable data[23:16]
B10	data_mask[1]		O	4	Byte enable data[15:08]
A11	data_mask[0]		O	4	Byte enable data {07:00]
A9	ns_ta_strb		I		Slow peripheral transfer acknowledge
A6	rw_n		O	4	Transfer direction
B7	clk_en[3]		O	4	SDRAM clock enable
D7	clk_en[2]		O	4	SDRAM clock enable
A7	clk_en[1]		O	4	SDRAM clock enable
B8	clk_en[0]		O	4	SDRAM clock enable
B4	cs[7]		O	4	Chip select 7, dy_cs3
A3	cs[6]		O	4	Chip select 6, st_cs3
A4	cs[5]		O	4	Chip select 5, dy_cs2
C5	cs[4]		O	4	Chip select 4, st_cs2
B5	cs[3]		O	4	Chip select 3, dy_cs1
B6	cs[2]		O	4	Chip select 2, st_cs1 (Flash boot)
D6	cs[1]		O	4	Chip select 1, dy_cs0 (Boot sdram)
C6	cs[0]		O	4	Chip select 0, st_cs0
C4	ras_n		O	4	SDRAM RAS
A2	cas_n		O	4	SDRAM CAS
C7	we_n		O	4	SDRAM write enable
B3	ap10		O	4	SDRAM A10(AP)
A8	st_oe_n		O	4	Static output enable

a. addr [27:24] reset to gpio mode. These address lines cannot be used for boot.
b. gpio[31:16] reset to memory data bus data [15:0].

Ethernet interface MAC

Pin	Signal	U/D	I/O	OD	Description
A12	mdc / gpio[32]	U	I/O	2	MII clock
D11	mdio / gpio[35]	U	I/O	2	MII data
B12	tx_clk / gpio[33]	U	I/O	2	TX clock
A16	txd[3] / gpio[47]	U	I/O	2	TX data 3

Downloaded from www.Manualslib.com manuals search engine

Pin	Signal	U/D	I/O	OD	Description
D12	txd[2] / gpio[46]	U	I/O	2	TX data 2
C12	txd[1] / gpio[45]	U	I/O	2	TX data 1
B13	txd[0] / gpio[44]	U	I/O	2	TX data 0
B15	tx_er / gpio[43]	U	I/O	2	TX code err
B14	tx_en / gpio[42]	U	I/O	2	TX enable
C14	col / gpio[48]	U	I/O	2	Collision
C13	crs / gpio[49]	U	I/O	2	Carrier sense
A14	rx_clk / gpio[34]	U	I/O	2	RX clock
E17	rxd[3] / gpio[41]	U	I/O	2	RX data 3
D16	rxd[2] / gpio[40]	U	I/O	2	RX data 2
B17	rxd[1] / gpio[39]	U	I/O	2	RX data 1
D13	rxd[0] / gpio[38]	U	I/O	2	RX data 0
C17	rx_er / gpio[37]	U	I/O	2	RX error
D17	rx_dv / gpio[36]	U	I/O	2	RX data valid

General purpose I/O (GPIO)

- Some signals are multiplexed to two or more GPIOs, to maximize the number of possible applications. These duplicate signals are marked as *(dup)* in the Descriptions column in the table. Selecting the primary GPIO pin and the duplicate GPIO pin for the same function is not recommended. If both the primary GPIO pin and duplicate GPIO pin are programmed for the same function, however, the primary GPIO pin has precedence and will be used.
- The I^2C module must be held in reset until the GPIO assigned to I^2C has been configured.

Downloaded from www.Manualslib.com manuals search engine

Note: All GPIOs except 12 and 16 to 31 are reset to mode 3, input. GPIO 12 is reset to mode 2, reset_done. GPIO 16 to 31 are reset to mode 0, external memory data 15:0.

Pin	Signal	U/D	I/O	OD	Description
K15	gpio[0]	U	I/O	2	0 DCD UART A 1 Ext DMA Done Ch 0 2 PIC_0_GEN_IO[0](I/O) 3 gpio[0] 4 SPI EN (dup)
K17	gpio[1]	U	I/O	2	0 CTS UART A 1 Ext Int 0 2 PIC_0_GEN_IO[1](I/O) 3 gpio[1] 4 Reserved
J17	gpio[2]	U	I/O	2	0 DSR UART A 1 Ext Int 1 2 PIC_0_GEN_IO[2](I/O) 3 gpio[2] 4 Reserved
J16	gpio[3]	U	I/O	2	0 RXD UART A 1 Ext DMA Pden Ch 0 2 PIC_0_GEN_IO[3](I/O) 3 gpio[3] 4 SPI RXD (dup)
H17	gpio[4]	U	I/O	2	0 RI UART A 1 Ext Int Ch 2 2 Ext Timer Event In Ch 6 3 gpio[4] 4 SPI CLK (dup)
H13	gpio[5]	U	I/O	2	0 RTS / RS485 Control UART A 1 Ext Int Ch 3 2 Ext Timer Event Out Ch 6 3 gpio[5] 4 SPI CLK (dup)
H14	gpio[6]	U	I/O	2	0 TXC / DTR UART A 1 Ext DMA Req Ch 0 2 Ext Timer Event In Ch 7 3 gpio[6] 4 PIC_DBG_DATA_OUT(O)
G14	gpio[7]	U	I/O	2	0 TXD UART A 1 Ext Timer Event In Ch 8 2 Ext Timer Event Out Ch 7 3 gpio[7] 4 SPI TXD (dup)

Downloaded from www.Manualslib.com manuals search engine

Pin	Signal	U/D	I/O	OD	Description
G17	gpio[8]	U	I/O	2	0 DCD / TXC UART C 1 Ext DMA Done Ch 1 2 Ext Timer Event Out Ch 8 3 gpio[8] 4 SPI EN (dup)
G15	gpio[9]	U	I/O	4	0 CTS UART C 1 I^2C SCL 2 Ext Int Ch 0 (dup) 3 gpio[9] 4 PIC_DBG_DATA_IN(I)
G16	gpio[10]	U	I/O	2	0 DSR UART C 1 QDC 1 2 Ext Int Ch 1 (dup) 3 gpio[10] 4 PIC_DBG_CLK(O)
F13	gpio[11]	U	I/O	2	0 RXD UART C 1 Ext DMA Pden Ch 1 2 Ext Int Ch 2 (dup) 3 gpio[11] 4 SPI RXD (boot)
F17	gpio[12]	U	I/O	4	0 RXC / RI UART C 1 I^2C SDA[a] 2 reset_done 3 gpio[12] 4 SPI CLK (dup)
F15	gpio[13]	U	I/O	2	0 RXC / RTS / RS485 Control UART C 1 QDC Q 2 Ext Timer Event Out Ch 9 3 gpio[13] 4 SPI CLK (boot)
E14	gpio[14]	U	I/O	2	0 TXC / DTR UART C 1 DMA Req Ch 1 2 PIC_0_CAN_RXD(I) 3 gpio[14] 4 SPI TXD (boot)
D14	gpio[15]	U	I/O	2	0 TXD UART C 1 Ext Timer Event In Ch 9 2 PIC_0_CAN_TXD(O) 3 gpio[15] 4 SPI EN (boot)

Downloaded from www.Manualslib.com manuals search engine

Pin	Signal	U/D	I/O	OD	Description
D3	gpio[16]	U	I/O	4	0 data[0] 1 DCD UART B 2 Ext Int Ch 0 (dup) 3 gpio[16]
B2	gpio[17]	U	I/O	4	0 data[1] 1 CTS UART B 2 Ext Int Ch 1 (dup) 3 gpio[17]
C2	gpio[18]	U	I/O	4	0 data[2] 1 DSR UART B 2 Ext Int Ch 2 (dup) 3 gpio[18]
D4	gpio[19]	U	I/O	4	0 data[3] 1 RXD UART B 2 EXT INT CH 3 (dup) 3 gpio[19]
B1	gpio[20]	U	I/O	4	0 data[4] 1 RI UART B 2 Ext DMA Done Ch 0 (dup) 3 gpio[20]
E3	gpio[21]	U	I/O	4	0 data[5] 1 RTS / RS485 Control UART B 2 Ext DMA Pden Ch 0 (dup) 3 gpio[21]
D2	gpio[22]	U	I/O	4	0 data[6] 1 TXC / DTR UART B 2 Ext DMA Done Ch 1 (dup) 3 gpio[22]
E4	gpio[23]	U	I/O	4	0 data[7] 1 TXD UART B 2 PIC_1_CAN_RXD(I) 3 gpio[23]
C1	gpio[24]	U	I/O	4	0 data[8] 1 DCD UART D 2 PIC_1_CAN_TXD(O) 3 gpio[24]
F5	gpio[25]	U	I/O	4	0 data[9] 1 CTS UART D 2 reset_done (dup) 3 gpio[25]

Downloaded from www.Manualslib.com manuals search engine

Pin	Signal	U/D	I/O	OD	Description
F4	gpio[26]	U	I/O	4	0 data[10] 1 DSR UART D 2 PIC_1_GEN_IO[0](I/O) 3 gpio[26]
F3	gpio[27]	U	I/O	4	0 data[11] 1 RXD UART D 2 PIC_1_GEN_IO[1](I/O) 3 gpio[27]
G5	gpio[28]	U	I/O	4	0 data[12] 1 RI UART D 2 PIC_1_GEN_IO[2](I/O) 3 gpio[28]
G4	gpio[29]	U	I/O	4	0 data[13] 1 RTS / RS485 Control UART D 2 PIC_1_GEN_IO[3](I/O) 3 gpio[29]
G3	gpio[30]	U	I/O	4	0 data[14] 1 TXC / DTR UART D 2 Reserved 3 gpio[30]
H4	gpio[31]	U	I/O	4	0 data[15] 1 TXD UART D 2 Reserved 3 gpio[31]
A12	gpio[32]	U	I/O	2	0 Ethernet MII MDC 1 PIC_0_GEN_IO[0](I/O)(dup) 2 Reserved 3 gpio[32]
B12	gpio[33]	U	I/O	2	0 Ethernet MII TXC 1 PIC_0_GEN_IO[1](I/O)(dup) 2 Reserved 3 gpio[33]
A14	gpio[34]	U	I/O	2	0 Ethernet MII RXC 1 PIC_0_GEN_IO[2](I/O)(dup) 2 Reserved 3 gpio[34]
D11	gpio[35]	U	I/O	2	0 Ethernet MII MDIO 1 PIC_0_GEN_IO[3](I/O)(dup) 2 Reserved 3 gpio[35]

Downloaded from www.Manualslib.com manuals search engine

Pin	Signal	U/D	I/O	OD	Description
D17	gpio[36]	U	I/O	2	0 Ethernet MII RX DV 1 PIC_0_GEN_IO[4](I/O)(dup) 2 Reserved 3 gpio[36]
C17	gpio[37]	U	I/O	2	0 Ethernet MII RX ER 1 PIC_0_GEN_IO[5](I/O)(dup) 2 Reserved 3 gpio[37]
D13	gpio[38]	U	I/O	2	0 Ethernet MII RXD[0] 1 PIC_0_GEN_IO[6](I/O)(dup) 2 Reserved 3 gpio[38]
B17	gpio[39]	U	I/O	2	0 Ethernet MII RXD[1] 1 PIC_0_GEN_IO[7](I/O)(dup) 2 Reserved 3 gpio[39]
D16	gpio[40]	U	I/O	2	0 Ethernet MII RXD [2] 1 PIC_1_GEN_IO[0](I/O)(dup) 2 Reserved 3 gpio[40]
E17	gpio[41]	U	I/O	2	0 Ethernet MII RXD[3] 1 PIC_1_GEN_IO[1](I/O)(dup) 2 Reserved 3 gpio[41]
B14	gpio[42]	U	I/O	2	0 Ethernet MII TX EN 1 PIC_1_GEN_IO[2](I/O)(dup) 2 Reserved 3 gpio[42]
B15	gpio[43]	U	I/O	2	0 Ethernet MII TX ER 1 PIC_1_GEN_IO[3](I/O)(dup) 2 Reserved 3 gpio[43]
B13	gpio[44]	U	I/O	2	) Ethernet MII TXD[0] 1 PIC_1_GEN_IO[4](I/O)(dup) 2 Reserved 3 gpio[44]
C12	gpio[45]	U	I/O	2	0 Ethernet MII TXD[1] 1 PIC_1_GEN_IO[5](I/O)(dup) 2 Reserved 3 gpio[45]

Downloaded from www.Manualslib.com manuals search engine

Pin	Signal	U/D	I/O	OD	Description
D12	gpio[46]	U	I/O	2	0 Ethernet MII TXD[2] 1 PIC_1_GEN_IO[6](I/O)(dup) 2 Reserved 3 gpio[46]
A16	gpio[47]	U	I/O	2	0 Ethernet MII TXD[3] 1 PIC_1_GEN_IO[7](I/O)(dup) 2 Reserved 3 gpio[47]
C14	gpio[48]	U	I/O	2	0 Ethernet MII COL 1 Reserved 2 Reserved 3 gpio[48]
C13	gpio[49]	U	I/O	2	0 Ethernet MII CRS 1 Reserved 2 Reserved 3 gpio[49]
C11	gpio[50]	U	I/O	2	0 Ethernet MII PHY Int 1 PIC_1_CLK(I) 2 PIC_1_CLK(O) 3 gpio[50]
E10	gpio[51]	U	I/O	2	0 DCD UART B (dup) 1 PIC_0_BUS_1[8](I/O) 2 PIC_1_BUS_1[8](I/O) 3 gpio[51]
D10	gpio[52]	U	I/O	2	0 CTS UART B (dup) 1 PIC_0_BUS_1[9](I/O) 2 PIC_1_BUS_1[9](I/O) 3 gpio[52]
C10	gpio[53]	U	I/O	2	0 DSR UART B (dup) 1 PIC_0_BUS_1[10](I/O) 2 PIC_1_BUS_1[10](I/O) 3 gpio[53]
C9	gpio[54]	U	I/O	2	0 RXD UART B (dup) 1 PIC_0_BUS_1[11](I/O) 2 PIC_1_BUS_1[11](I/O) 3 gpio[54]
H5	gpio[55]	U	I/O	2	0 RI UART B (dup) 1 PIC_0_BUS_1[12](I/O) 2 PIC_1_BUS_1[12](I/O) 3 gpio[55]

Downloaded from www.Manualslib.com manuals search engine

Pin	Signal	U/D	I/O	OD	Description
J4	gpio[56]	U	I/O	2	0 RTS/RS485 Control UART B (dup) 1 PIC_0_BUS_1[13](I/O) 2 PIC_1_BUS_1[13](I/O) 3 gpio[56]
K3	gpio[57]	U	I/O	2	0 TXC/DTR UART B (dup) 1 PIC_0_BUS_1[14](I/O) 2 PIC_1_BUS_1[14](I/O) 3 gpio[57]
K4	gpio[58]	U	I/O	2	0 TXD UART B (dup) 1 PIC_0_BUS_1[15](I/O) 2 PIC_1_BUS_1[15](I/O) 3 gpio[58]
K5	gpio[59]	U	I/O	2	0 DCD UART D (dup) 1 PIC_0_BUS_1[16](I/O) 2 PIC_1_BUS_1[16](I/O) 3 gpio[59]
R6	gpio[60]	U	I/O	2	0 CTS UART D (dup) 1 PIC_0_BUS_1[17](I/O) 2 PIC_1_BUS_1[17](I/O) 3 gpio[60]
P6	gpio[61]	U	I/O	2	0 DSR UART D (dup) 1 PIC_0_BUS_1[18](I/O) 2 PIC_1_BUS_1[18](I/O) 3 gpio[61]
R7	gpio[62]	U	I/O	2	0 RXD UART D (dup) 1 PIC_0_BUS_1[19](I/O) 2 PIC_1_BUS_1[19](I/O) 3 gpio[62]
P7	gpio[63]	U	I/O	2	0 RI UART D (dup) 1 PIC_0_BUS_1[20](I/O) 2 PIC_1_BUS_1[20](I/O) 3 gpio[63]
R8	gpio[64]	U	I/O	2	0 RTS/R5485 Control UART D (dup) 1 PIC_0_BUS_1[21](I/O) 2 PIC_1_BUS_1[21](I/O) 3 gpio[64]
P8	gpio[65]	U	I/O	2	0 TXC/DTR UART D (dup) 1 PIC_0_BUS_1[22](I/O) 2 PIC_1_BUS_1[22](I/O) 3 gpio[65]

Downloaded from www.Manualslib.com manuals search engine

Pin	Signal	U/D	I/O	OD	Description
N8	gpio[66]	U	I/O	2	0 TXD UART D (dup) 1 PIC_0_BUS_1[23](I/O) 2 PIC_1_BUS_1[23](I/O) 3 gpio[66]
P9	gpio[67]	U	I/O	2	0 PIC_0_CLK(I) 1 PIC_0_CLK(O) 2 Ext Int Ch 3 (dup) 3 gpio[67]
R10	gpio[68]	U	I/O	2	0 PIC_0_GEN_IO[0](I/O)(dup) 1 PIC_1_GEN_IO[0](I/O) 2 PIC_1_CAN_RXD(I)(dup) 3 gpio[68]
P10	gpio[69]	U	I/O	2	0 PIC_0_GEN_IO[1](I/O)(dup) 1 PIC_1_GEN_IO[1](I/O) 2 PIC_1_CAN_TXD(O)(dup) 3 gpio[69]
N10	gpio[70]	U	I/O	2	0 PIC_0_GEN_IO[2](I/O)(dup) 1 PIC_1_GEN_IO[2](I/O) 2 PWM Ch 0 3 gpio[70]
P11	gpio[71]	U	I/O	2	0 PIC_0_GEN_IO[3](I/O)(dup) 1 PIC_1_GEN_IO[3](I/O) 2 PWM Ch 1 3 gpio[71]
N12	gpio[72]	U	I/O	2	0 PIC_0_GEN_IO[4](I/O) 1 PIC_1_GEN_IO[4](I/O) 2 PWM Ch 2 3 gpio[72]
R13	gpio[73]	U	I/O	2	0 PIC_0_GEN_IO[5](I/O) 1 PIC_1_GEN_IO[5](I/O) 2 PWM Ch 3 3 gpio[73]
P13	gpio[74]	U	I/O	2	0 PIC_0_GEN_IO[6](I/O) 1 PIC_0_GEN_IO[6](I/O) 2 Ext Timer Event In Ch 0 3 gpio[74]
U16	gpio[75]	U	I/O	2	0 PIC_0_GEN_IO[7](I/O) 1 PIC_1_GEN_IO[7](I/O) 2 Ext Timer Event in Ch 1 3 gpio[75]

Downloaded from www.Manualslib.com manuals search engine

Pin	Signal	U/D	I/O	OD	Description
T15	gpio[76]	U	I/O	2	0 PIC_0_CTL_IO[0](I/O) 1 PIC_1_CTL_IO[0](I/O) 2 Ext Timer Event in Ch 2 3 gpio[76]
T16	gpio[77]	U	I/O	2	0 PIC_0_CTL_IO[1](I/O) 1 PIC_1_CTL_IO[1](I/O) 2 Ext Timer Event in Ch 3 3 gpio[77]
R14	gpio[78]	U	I.O	2	0 PIC_0_CTL_IO[2](I/O) 1 PIC_1_CTL_IO[2](I/O) 2 Ext Timer Event in Ch 4 3 gpio[78]
P14	gpio[79]	U	I/O	2	0 PIC_0_CTL_IO[3](I/O) 1 PIC_1_CTL_IO[3](I/O) 2 Ext Timer Event in Ch 5 3 gpio[79]
R17	gpio[80]	U	I/O	2	0 PIC_0_BUS_0[0](I/O) 1 PIC_1_BUS_0[0](I/O) 2 Ext Timer Event in Ch 6 (dup) 3 gpio[80]
P17	gpio[81]	U	I/O	2	0 PIC_0_BUS_0[1](I/O) 1 PIC_1_BUS_0[1](I/O) 2 Ext Timer Event in Ch 7(dup) 3 gpio[81]
N16	gpio[82]	U	I/O	2	0 PIC_0_BUS_0[2](I/O) 1 PIC_1_BUS_0[2](I/O) 2 Ext Timer Event in Ch 8 (dup) 3 gpio[82]
N17	gpio[83]	U	I/O	2	0 PIC_0_BUS_0[3](I/O) 1 PIC_1_BUS_0[3](I/O) 2 Ext Timer Event in Ch 9 (dup) 3 gpio[83]
M17	gpio[84]	U	I/O	2	0 PIC_0_BUS_0[4](I/O) 1 PIC_1_BUS_0[4](I/O) 2 Ext Timer Event Out Ch 0 3 gpio[84]
L15	gpio[85]	U	I/O	2	0 PIC_0_BUS_0[5](I/O) 1 PIC_1_BUS_0[5](I/O) 2 Ext Timer Event Out Ch 1 3 gpio[85]

Downloaded from www.Manualslib.com manuals search engine

Pin	Signal	U/D	I/O	OD	Description
K13	gpio[86]	U	I/O	2	0 PIC_0_BUS_0[6](I/O) 1 PIC_1_BUS_0[6](I/O) 2 Ext Timer Event Out Ch 2 3 gpio[86]
K16	gpio[87]	U	I/O	2	0 PIC_0_BUS_0[7](I/O) 1 PIC_1_BUS_0[7](I/O) 2 Ext Timer Event Out Ch 3 3 gpio[87]
K14	gpio[88]	U	I/O	2	0 PIC_0_BUS_0[8](I/O) 1 PIC_1_BUS_0[8](I/O) 2 Ext Timer Event Out Ch 4 3 gpio[88]
J14	gpio[89]	U	I/O	2	0 PIC_0_BUS_0[9](I/O) 1 PIC_1_BUS_0[9](I/O) 2 Ext Timer Event Out Ch 5 3 gpio[89]
H16	gpio[90]	U	I/O	2	0 PIC_0_BUS_0[10](I/O) 1 PIC_1_BUS_0[10](I/O) 2 Ext Timer Event Out Ch 6 3 gpio[90]
H15	gpio[91]	U	I/O	2	0 PIC_0_BUS_0[11](I/O) 1 PIC_1_BUS_0[11](I/O) 2 Ext Timer Event Out Ch 7 3 gpio[91]
F14	gpio[92]	U	I/O	2	0 PIC_0_BUS_0[12](I/O) 1 PIC_1_BUS_0[12](I/O) 2 Ext Timer Event Out Ch 8 3 gpio[92]
F16	gpio[93]	U	I/O	2	0 PIC_0_BUS_0[13](I/O) 1 PIC_1_BUS_0[13](I/O) 2 Ext Timer Event Out Ch 9 3 gpio[92]
E15	gpio[94]	U	I/O	2	0 PIC_0_BUS_0[14](I/O) 1 PIC_1_BUS_0[14](I/O) 2 QDC I (dup) 3 gpio[94]
E16	gpio[95]	U	I/O	2	0 PIC_0_BUS_0[15](I/O) 1 PIC_1_BUS_0[15](I/O) 2 QDC Q (dup) 3 gpio[95]

Downloaded from www.Manualslib.com manuals search engine

Pin	Signal	U/D	I/O	OD	Description
C16	gpio[96]	U	I/O	2	0 PIC_0_BUS_1[0](I/O) 1 PIC_1_BUS_1[0](I/O) 2 PIC_0_CAN_RXD(I)(dup) 3 gpio[96]
B16	gpio[97]	U	I/O	2	0 PIC_0_BUS_1[1](I/O) 1 PIC_1_BUS_1[1](I/O) 2 PIC_0_CAN_TXD(O)(dup) 3 gpio97
D15	gpio[98]	U	I/O	2	0 PIC_0_BUS_1[2](I/O) 1 PIC_1_BUS_1[2](I/O) 2 PIC_1_CAN_RXD(I)(dup) 3 gpio[98]
E8	gpio[99]	U	I/O	2	0 PIC_0_BUS_1[3](I/O) 1 PIC_1_BUS_1[3](I/O) 2 PIC_1_CAN_TXD(O)(dup) 3 gpio[99]
D8	gpio[100]	U	I/O	2	0 PIC_0_BUS_1[4](I/O) 1 PIC_1_BUS_1[4](I/O) 2 PWM Ch 4 3 gpio[100]
C8	gpio[101]	U	I/O	2	0 PIC_0_BUS_1[5](I/O) 1 PIC_1_BUS_1[5](I/O) 2 Ext Int Ch 3 (dup) 3 gpio[101]
E6	gpio[102]	U	I/O	4	0 PIC_0_BUS_1[6](I/O) 1 PIC_1_BUS_1[6](I/O) 2 I^2C SCL (dup) 3 gpio[102]
D5	gpio[103]	U	I/O	4	0 PIC_0_BUS_1[7](I/O) 1 PIC_1_BUS_1[7](I/O) 2 I^2C SDA (dup) 3 gpio[103]
R12	gpio_a[0]	U	I/O	4	0 addr[24] 1 I^2C SCL (dup) 2 Ext Int Ch 0 (dup) 3 gpio_a[0], Boot width[1]
U15	gpio_a[1]	U	I/O	4	0 addr[25] 1 I^2C SDA (dup) 2 Ext Int Ch 1(dup) 3 Ext Int Ch 0]

Downloaded from www.Manualslib.com manuals search engine

Pin	Signal	U/D	I/O	OD	Description
T14	gpio_a[2]	U	I/O	4	0 addr[26] 1 Reserved 1 cs0_we_n 2 Ext Int Ch 2 (dup) 3 gpio_a[2], SPI boot
P12	gpio_a[3]	U	I/O	4	0 addr[27] 1 Reserved 1 cs0_oe_n 2 UART ref clock 3 gpio_a[3], Endian

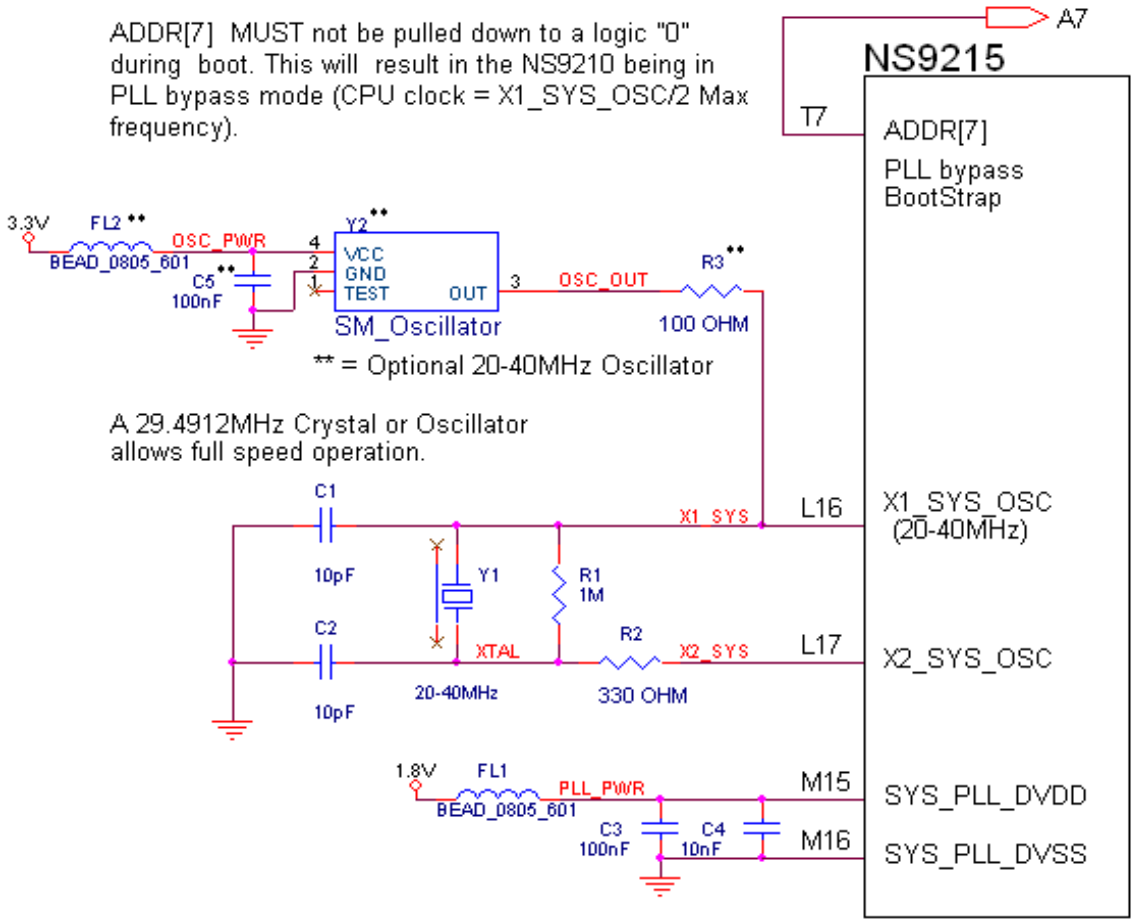
a. There is a possible conflict when gpio12 is used as the I2C_SDA signal. in this case the I2C_SDA signal is driven low while in reset, then driven active high after end of reset, until software configures this pin for the I2C function.

System clock

Pin	Signal	U/D	I/O	OD	Description
L16	x1_sys_osc		I		System oscillator circuit in
L17	x2_sys_osc		O		System oscillator circuit out
M15	sys_pll_dvdd		P		PLL clean power
M16	sys_pll_dvss		P		PLL clean ground
P2	x1_rtc_osc		I		RTC oscillator circuit in (32.768 KHz)
R2	x2_rtc_osc		O		RTC oscillator circuit out

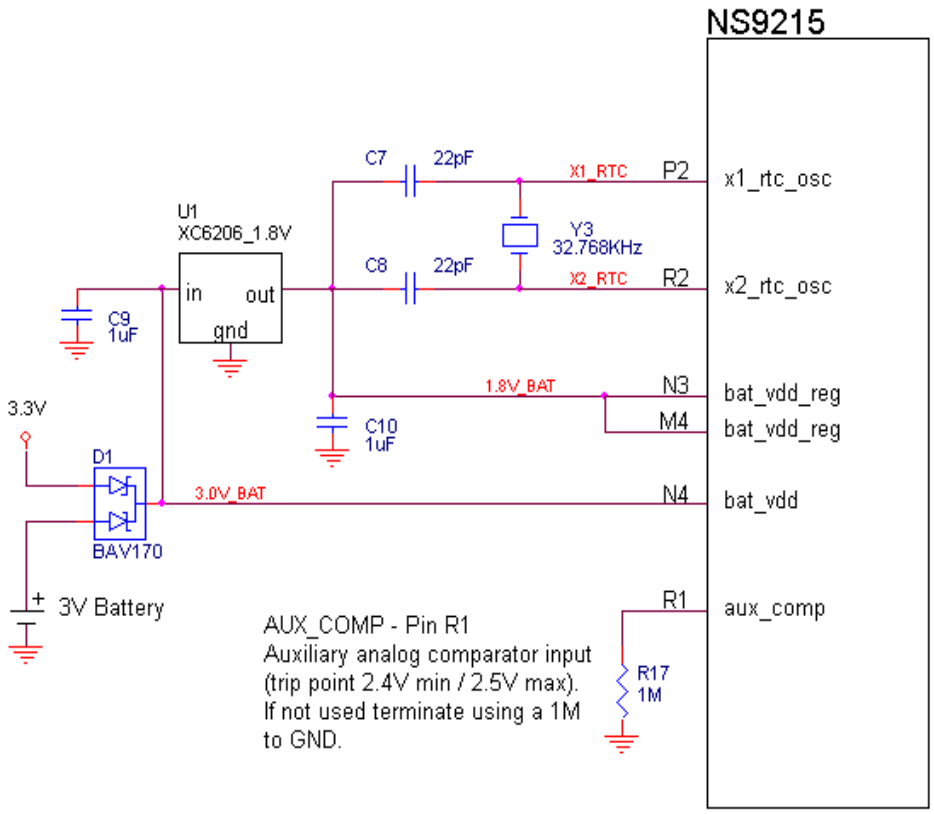
Downloaded from www.Manualslib.com manuals search engine

System clock drawing



Downloaded from www.Manualslib.com manuals search engine

RTC clock and battery backup drawing



Note: If RTC battery backup is not used, the following connection changes can be made.

N3, M4	bat_vdd_reg	tie to 1.8V
32.788kHz	crystal load capacitors	tie to N3, M4 (1.8V)
N4	bat_vdd	tie to 3.3V
R1	aux_comp	tie to ground

System mode

Pin	Signal	U/D	I/O	OD	Description
M13	sys_mode_2		I		test mode pins
M14	sys_mode_1		I		test mode pins
L14	sys_mode_0		I		test mode pins

Downloaded from www.Manualslib.com manuals search engine

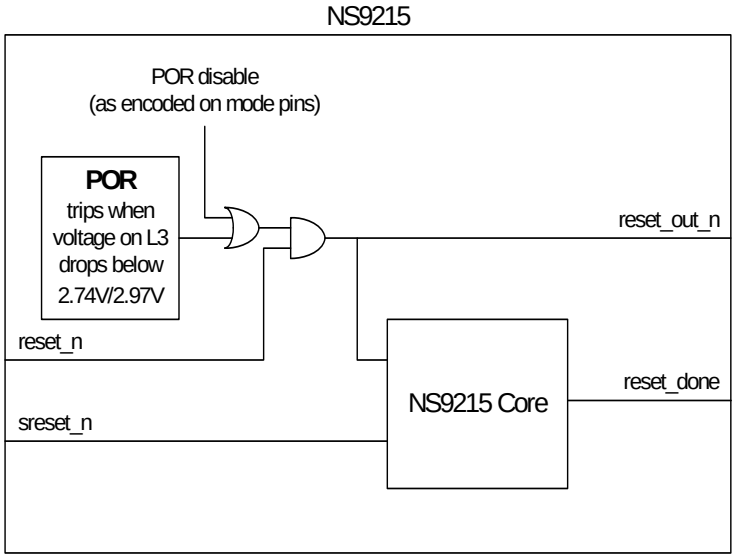
sys_mode_2	sys_mode_1	sys_mode_0	Description
0	0	0	manufacturing test
0	0	1	manufacturing test
0	1	0	manufacturing test
0	1	1	normal operation, boundary scan enabled, POR disabled
1	0	0	normal operation, boundary scan enabled, POR enabled
1	0	1	board test mode, all outputs tristated
1	1	0	normal operation, ARM debug enabled, POR disabled
1	1	1	normal operation, ARM debug enabled, POR enabled

Downloaded from www.Manualslib.com manuals search engine

System reset

Pin	Signal	U/D	I/O	OD	Description
E12	reset_n	U	I		System reset
A5	reset_out_n		O	2	System reset output
A13	reset_done		O	2	Reset done
D9	sreset_n	U	I		Soft system reset

	RESET_n pin	SRESET_n pin	PLL Config Reg. Update	Watchdog Time-Out Reset
SPI	YES	YES	YES	YES
BootStrapping PL	YES	NO	NO	NO
Other Strappings (Endianess	YES	NO	NO	NO
GPIO Configuration	YES	NO	NO	NO
Other (ASIC) Registers	YES	YES	YES	YEs

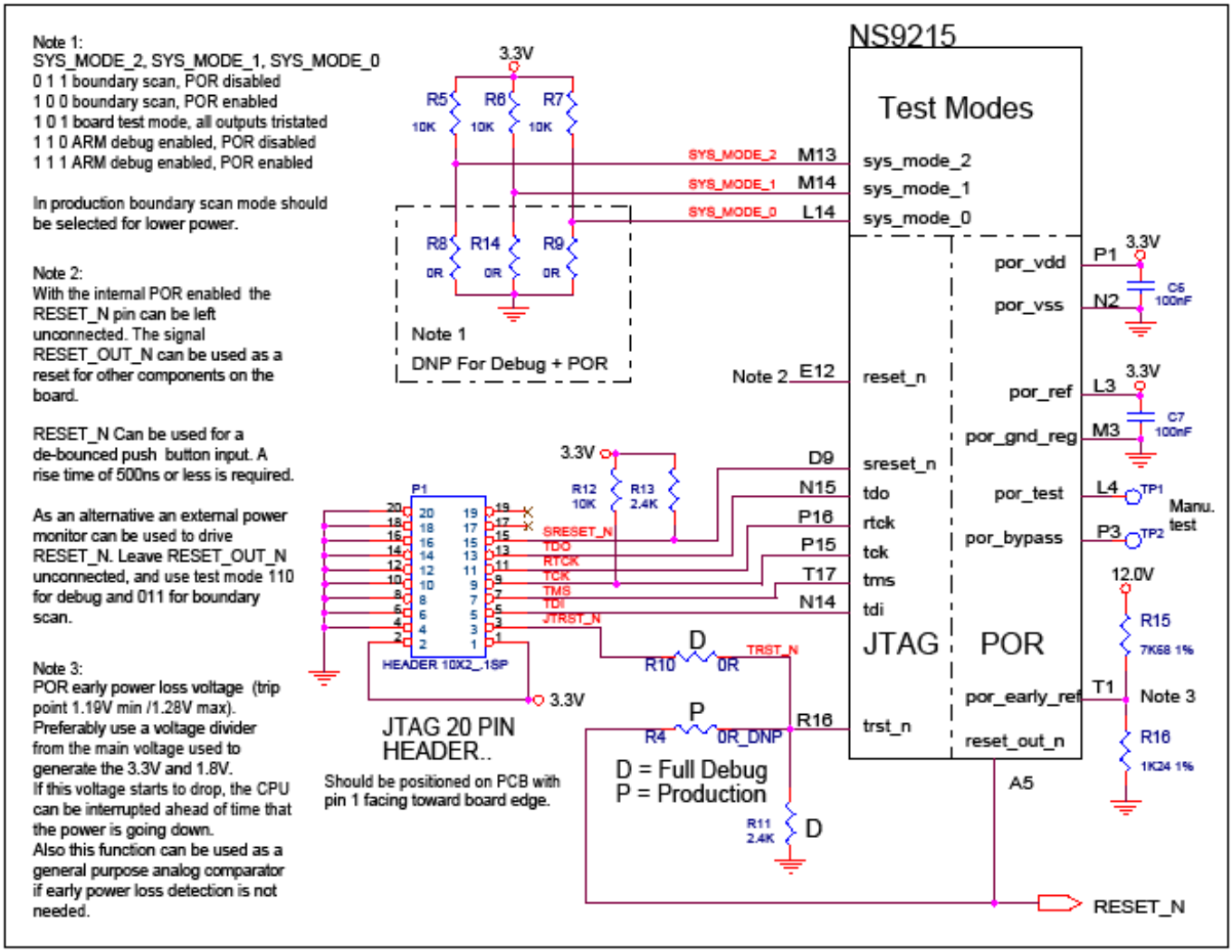


Definitions
reset_n – hardware reset input buffer with pull-up resistor
sreset_n – soft reset input buffer with pull-up resistor, does not reset the PLL
reset_out_n – hardware reset to NS9215 core and output buffer, resets all logic in NS9215 core including PLL
reset_done – reflects the state of the ARM926 reset, for any type of reset event

Downloaded from www.Manualslib.com manuals search engine

JTAG Test

Pin	Signal	U/D	I/O	OD	Description
N14	tdi	U	I		Test data in
N15	tdo		O	2	Test data out
T17	tms	U	I		Test mode select
R16	trst_n	U	I		Test mode reset. For normal operation, this pin is tied to ground or pulled down.
P15	tck		I		Test mode clock
P16	rtck		O	2	Test mode return clock



Downloaded from www.Manualslib.com manuals search engine

ADC

Pin	Signal	U/D	I/O	OD	Description
P4	agnd_ref_adc				Analog reference ground
P5	VREF_adc				Analog reference voltage (3.3max
T2	vss_adc				ADC_VSS
N6	vdd_adc				ADC VDD (3.3V)
R4	vin0_adc		I		ADC input 0
T3	vin1_adc		I		ADC input 1
R5	vin2_adc		I		ADC input 2
U2	vin3_adc		I		ADC input 3
T4	vin4_adc		I		ADC input 4
U3	vin5_adc		I		ADC input 5
T5	vin6_adc		I		ADC input 6
U4	vin7_adc		I		ADC input 7

If the ADC feature is not used, the inputs must be terminated as shown below:

P4	agnd_ref_adc	tie to ground
P5	VREF_adc	tie to ground
T2	vss_adc	tie to ground
N6	vdd_adc	tie to 3.3V
R4	vin0_adc	tie to ground
T3	vin1_adc	tie to ground
R5	vin2_adc	tie to ground
U2	vin3_adc	tie to ground
T4	vin4_adc	tie to ground
U3	vin5_adc	tie to ground
T5	vin6_adc	tie to ground
U4	vin7_adc	tie to ground

Downloaded from www.Manualslib.com manuals search engine

POR and battery-backed logic

Pin	Signal	U/D	I/O	OD	Description
M3	por_gnd_reg				POR reference ground
N2	por_vss				POR VSS
P1	por_vdd				POR VDD (3.3V)
L3	por_reference				POR reference trip voltage (2.74V min / 2.97V max)
T1	por_early_reference				POR early power loss voltage (1.19V min / 1.28V max)
N4	bat_vdd				Battery VDD (3.0V)
R1	aux_comp				Auxiliary analog comparator input (trip point 2.4V min / 2.5V max)
N3, M4	bat_vdd_reg				Battery regulated core VDD (1.8V)
P3	por_bypass	U	I		POR bypass, pull low to disable POR
L4	por_test				POR analog test pin, leave unconnected

The POR will generate keep reset_out_n low between 75ms and 300ms after 3.3V reaches the POR reference trip voltage threshold. The POR reference trip voltage is between 2.74V and 2.97V, with hysteresis between 50mV and 80mV.

If the POR feature is not used, and the RTC is used, the inputs must be terminated as shown below.

M3	por_gnd_reg	tie to ground
N2	por_vss	tie to ground
P1	por_vdd	tie to 3.3V
L3	por_reference	tie to 3.3V
T1	por_early_reference	tie to ground
P3	por_bypass	tie to 1.8V
E12	reset_n	tie to system reset (remains active low 40 ms Min. after 3.3V & 1.8V are valid)
A5	reset_out_n	leave open
M13, M14, L14	sys_mode [2.0]	POR disabled (See System mode table & JTAG drawing following JTAG Test table)

Downloaded from www.Manualslib.com manuals search engine

If the RTC feature is not used, the inputs must be terminated as shown below.

N4	Bat_vdd	tie to 3.3V
R1	aux_comp	tie to ground
N3, M4	bat_vdd_reg	tie to ground
P2	x1_rtc_osc	tie to ground
R2	x2_rtc_osc	leave open

If the RTC feature is used, see RTC clock and battery backup drawing on page 45.

Power and ground

Pin	Signal
E7, E11, G7, G11, G13, L5, L7, L11, L13, N7, N11	Core VCC (1.8V)
A1. A17. C3, C15, E5, E9, E13, J5, J13, J15, N5, N9, N13, R3, R15, U1, U17	I/O VCC (3.3V)
G8, G9, G10, H7, H8, H9, H10, H11, J7, J8, J9, J10, J11, K7, K8, K9, K10, K11, L8, L9, L10, M5	GND

Downloaded from www.Manualslib.com manuals search engine

Downloaded from www.Manualslib.com manuals search engine

I/O Control Module

CHAPTER 2

The NS9215 ASIC contains 108 pins that are designated as general purpose I/O (GPIO).

- The first 16 GPIO can be configured to serve one of five functions.
- The remaining GPIO can be configured to serve one of four functions.

All signals set to a disabled peripheral are held in the inactive state. The I/O control module contains the control register and multiplexing logic required to accomplish this task.

System memory bus I/O control

The registers in this section control these system memory I/O configuration options:

- System chip select options, used to select which chip select is output
- Upper address option

Control and Status registers

The I/O control module configuration registers are located at base address 0xA090_2000.

Register address map

Address	Description	Access	Reset value
A090_2000	GPIO Configuration Register #0	R/W	0x18181818
A090_2004	GPIO Configuration Register #1	R/W	0x18181818
A090_2008	GPIO Configuration Register #2	R/W	0x18181818

Downloaded from www.Manualslib.com manuals search engine

Address	Description	Access	Reset value
A090_200C	GPIO Configuration Register #3	R/W	0x18181810
A090_2010	GPIO Configuration Register #4	R/W	0x00000000
A090_2014	GPIO Configuration Register #5	R/W	0x00000000
A090_2018	GPIO Configuration Register #6	R/W	0x00000000
A090_201C	GPIO Configuration Register #7	R/W	0x00000000
A090_2020	GPIO Configuration Register #8	R/W	0x18181818
A090_2024	GPIO Configuration Register #9	R/W	0x18181818
A090_2028	GPIO Configuration Register #10	R/W	0x18181818
A090_202C	GPIO Configuration Register #11	R/W	0x18181818
A090_2030	GPIO Configuration Register #12	R/W	0x18181818
A090_2034	GPIO Configuration Register #13	R/W	0x18181818
A090_2038	GPIO Configuration Register #14	R/W	0x18181818
A090_203C	GPIO Configuration Register #15	R/W	0x18181818
A090_2040	GPIO Configuration Register #16	R/W	0x18181818
A090_2044	GPIO Configuration Register #17	R/W	0x18181818
A090_2048	GPIO Configuration Register #18	R/W	0x18181818
A090_204C	GPIO Configuration Register #19	R/W	0x18181818
A090_2050	GPIO Configuration Register #20	R/W	0x18181818
A090_2054	GPIO Configuration Register #21	R/W	0x18181818
A090_2058	GPIO Configuration Register #22	R/W	0x18181818
A090_205C	GPIO Configuration Register #23	R/W	0x18181818
A090_2060	GPIO Configuration Register #24	R/W	0x18181818
A090_2064	GPIO Configuration Register #25	R/W	0x18181818
A090_2068	GPIO Configuration Register #26	R/W	0x18181818
A090_206C	GPIO Control Register #0	R/W	0x00000000
A090_2070	GPIO Control Register #1	R/W	0x00000000
A090_2074	GPIO Control Register #2	R/W	0x00000000
A090_2078	GPIO Control Register #3	R/W	0x00000000
A090_207C	GPIO Status Register #0	R	Undefined[1]
A090_2080	GPIO Status Register #1	R	Undefined[1]
A090_2084	GPIO Status Register #2	R	Undefined[1]
A090_2088	GPIO Status Register #3	R	Undefined[1]

Downloaded from www.Manualslib.com manuals search engine

Address	Description	Access	Reset value
A090_208C	Memory Bus Configuration register	R/W	007D6344

1 The reset values for all the status bits are undefined because they depend on the state of the GPIO pins to NS9215.

GPIO Configuration registers

GPIO Configuration registers #0 through #26 contain the configuration information for each of the 108 GPIO pins. Each GPIO pin can have up to four functions. Configure each pin for the function and direction needed, using the configuration options shown below.

GPIO configuration options

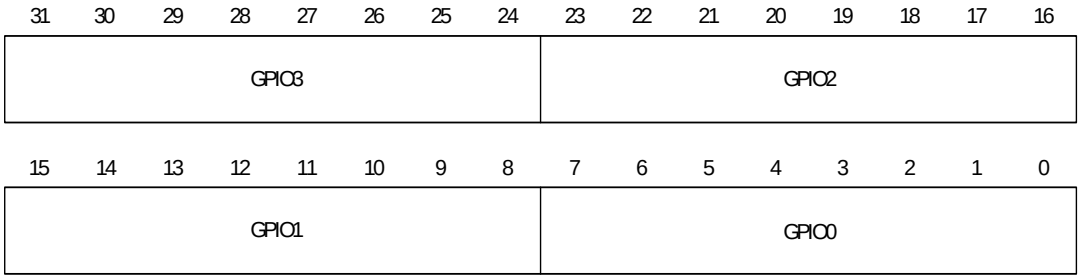
Each GPIO configuration section is set up the same way. This table shows the settings using bits D07:00; the same settings apply to the corresponding bits in D15:08, D23:D16, and D31:24.

Bit(s)	Mnemonic	Description
D07:06	Reserved	N/A
D05:03	FUNC	Use these bits to select the function you want to use. For a definition of each function, see “General purpose I/O (GPIO)” on page 31. 000 Function #0 001 Function #1 010 Function #2 011 Function #3 100 Function #4 (applicable only for GPIO 0–15)
D02	DIR	Controls the pin direction when the FUNC field is configured for GPIO mode, function #3. 0 Input 1 Output All GPIO pins reset to the input state. Note: The pin direction is controlled by the selected function in modes #0 through #2.
D01	INV	Controls the inversion function of the GPIO pin. 0 Disables the inversion function 1 Enables the inversion function This bit applies to all functional modes.
D00	PUDIS	Controls the GPIO pin pullup resistor operation. 0 Enables the pullup 1 Disables the pullup Note: The pullup cannot be disabled on GPIO[9], GPIO[12], and on GPIO_A[0] and GPIO_A[1].

Downloaded from www.Manualslib.com manuals search engine

GPIO Configuration Register #0

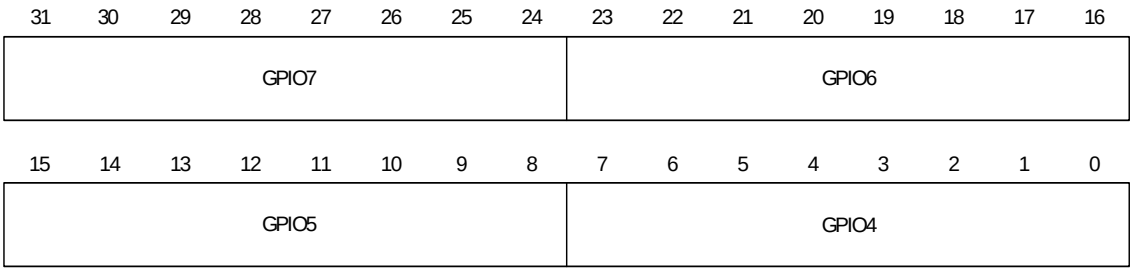
Address: A090_2000



Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO3	0x18	GPIO[3] configuration
D23:16	R/W	GPIO2	0x18	GPIO[2] configuration
D15:08	R/W	GPIO1	0x18	GPIO[1] configuration
D07:00	R/W	GPIO0	0x18	GPIO[0] configuration

GPIO Configuration Register #1

Address: A090_2004

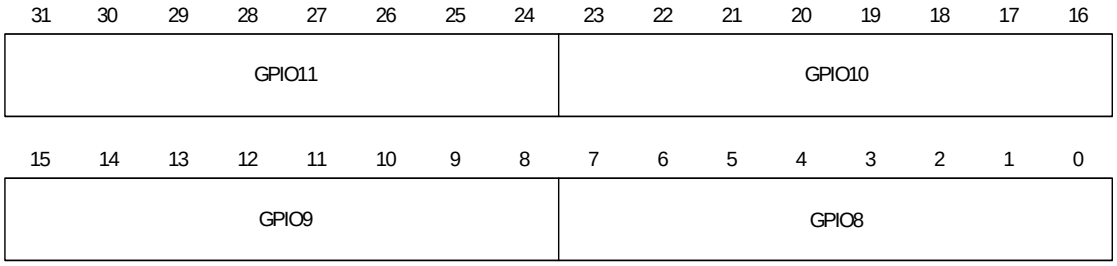


Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO7	0x18	GPIO[7] configuration
D23:16	R/W	GPIO6	0x18	GPIO[6] configuration
D15:08	R/W	GPIO5	0x18	GPIO[5] configuration
D07:00	R/W	GPIO4	0x18	GPIO[4] configuration

Downloaded from www.Manualslib.com manuals search engine

GPIO Configuration Register #2

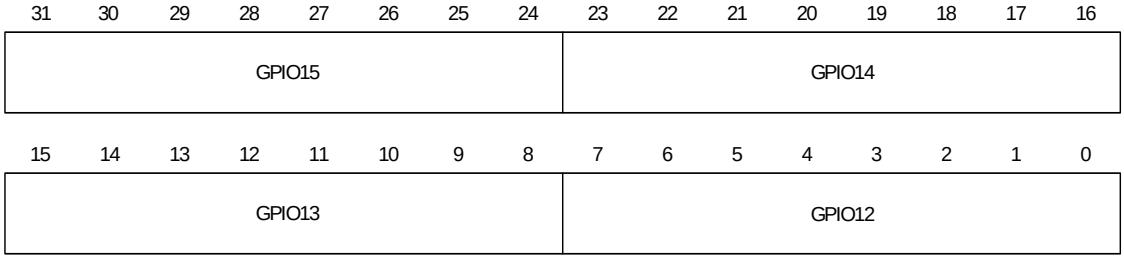
Address: A090_2008



Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO11	0x18	GPIO[11] configuration
D23:16	R/W	GPIO10	0x18	GPIO[10] configuration
D15:08	R/W	GPIO9	0x18	GPIO[9] configuration
D07:00	R/W	GPIO8	0x18	GPIO[8] configuration

GPIO Configuration Register #3

Address: A090_200C

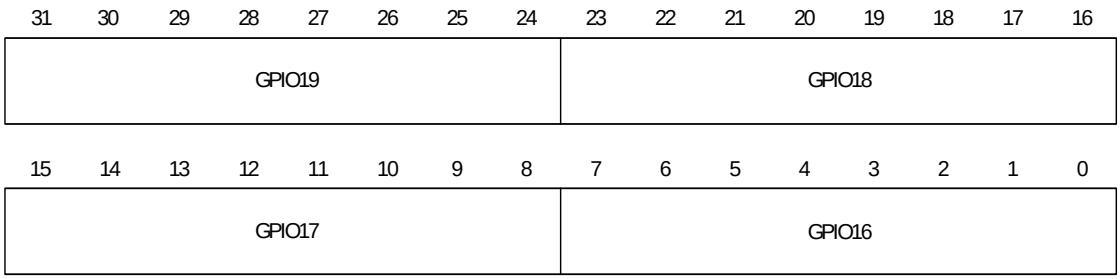


Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO15	0x18	GPIO[15] configuration
D23:16	R/W	GPIO14	0x18	GPIO[14 configuration
D15:08	R/W	GPIO13	0x18	GPIO[13] configuration
D07:00	R/W	GPIO12	0x10	GPIO[12] configuration

Downloaded from www.Manualslib.com manuals search engine

GPIO Configuration Register #4

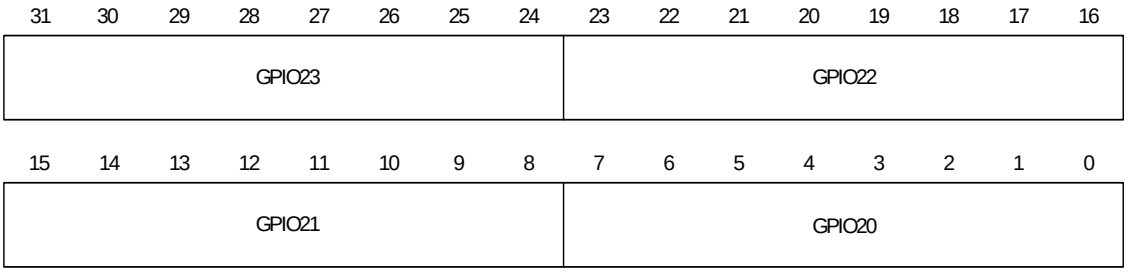
Address: A090_2010



Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO19	0x00	GPIO[19] configuration
D23:16	R/W	GPIO18	0x00	GPIO[18] configuration
D15:08	R/W	GPIO17	0x00	GPIO[17] configuration
D07:00	R/W	GPIO16	0x00	GPIO[16] configuration

GPIO Configuration Register #5

Address: A090_2014

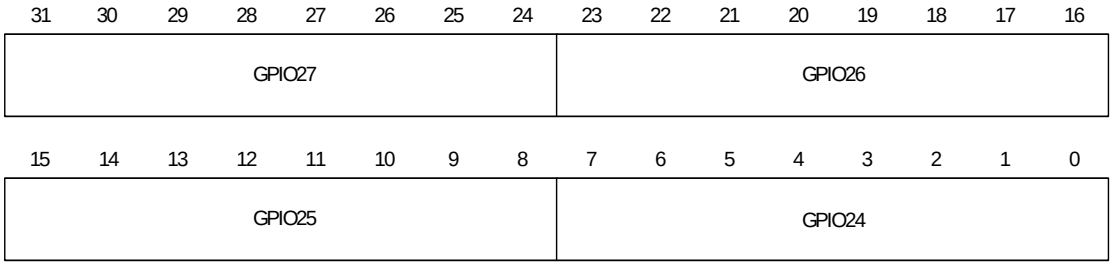


Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO23	0x00	GPIO[23] configuration
D23:16	R/W	GPIO22	0x00	GPIO[22] configuration
D15:08	R/W	GPIO21	0x00	GPIO[21] configuration
D07:00	R/W	GPIO20	0x00	GPIO[20] configuration

Downloaded from www.Manualslib.com manuals search engine

GPIO Configuration Register #6

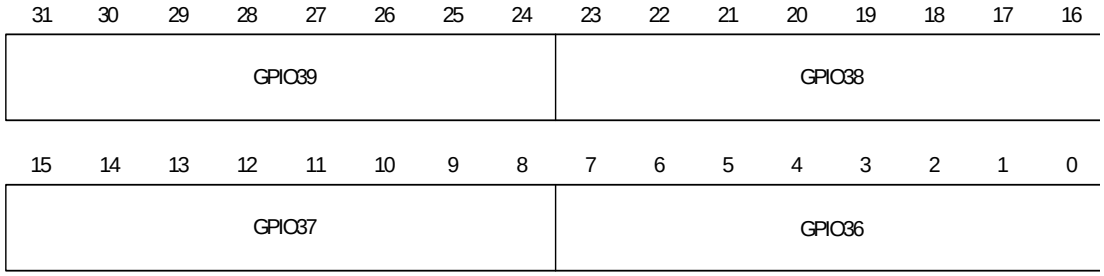
Address: A090_2018



Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO27	0x00	GPIO[27] configuration
D23:16	R/W	GPIO26	0x00	GPIO[26] configuration
D15:08	R/W	GPIO25	0x00	GPIO[25] configuration
D07:00	R/W	GPIO24	0x00	GPIO[24] configuration

GPIO Configuration Register #7

Address: A090_201C

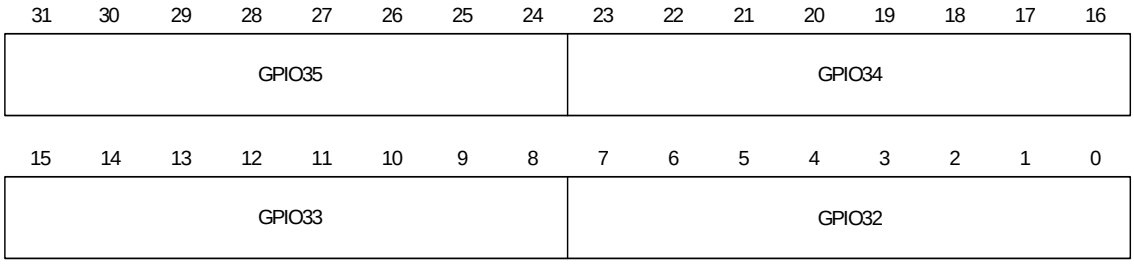


Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO31	0x00	GPIO[31] configuration
D23:16	R/W	GPIO30	0x00	GPIO[30] configuration
D15:08	R/W	GPIO29	0x00	GPIO[29] configuration
D07:00	R/W	GPIO28	0x00	GPIO[28] configuration

Downloaded from www.Manualslib.com manuals search engine

GPIO Configuration Register #8

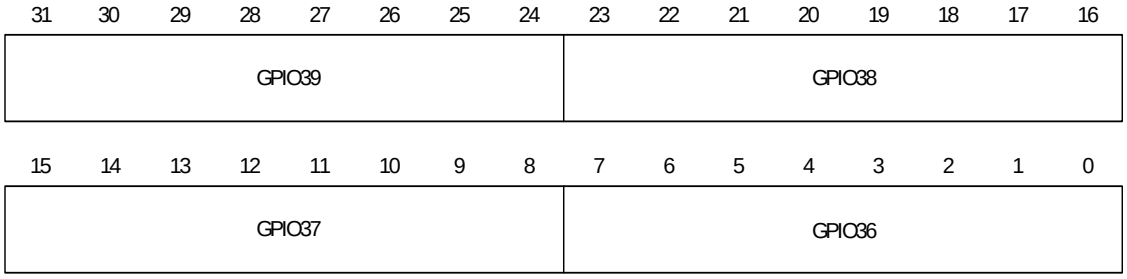
Address: A090_2020



Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO35	0x18	GPIO[35] configuration
D23:16	R/W	GPIO34	0x18	GPIO[34] configuration
D15:08	R/W	GPIO33	0x18	GPIO[33] configuration
D07:00	R/W	GPIO32	0x18	GPIO[32] configuration

GPIO Configuration Register #9

Address: A090_2024

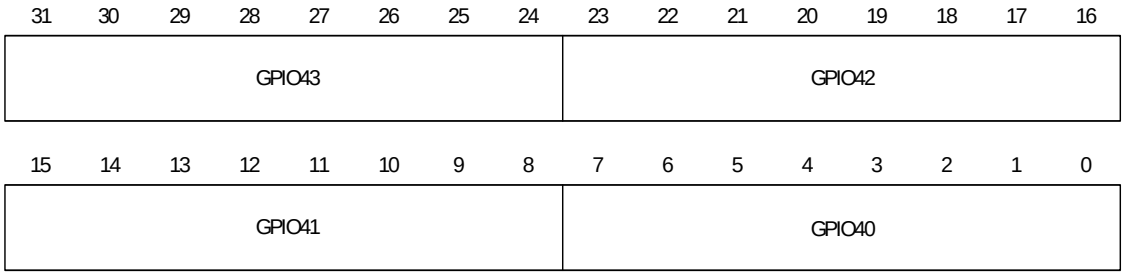


Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO39	0x18	GPIO[39] configuration
D23:16	R/W	GPIO38	0x18	GPIO[38] configuration
D15:08	R/W	GPIO37	0x18	GPIO[37] configuration
D07:00	R/W	GPIO36	0x18	GPIO[36] configuration

Downloaded from www.Manualslib.com manuals search engine

GPIO Configuration Register #10

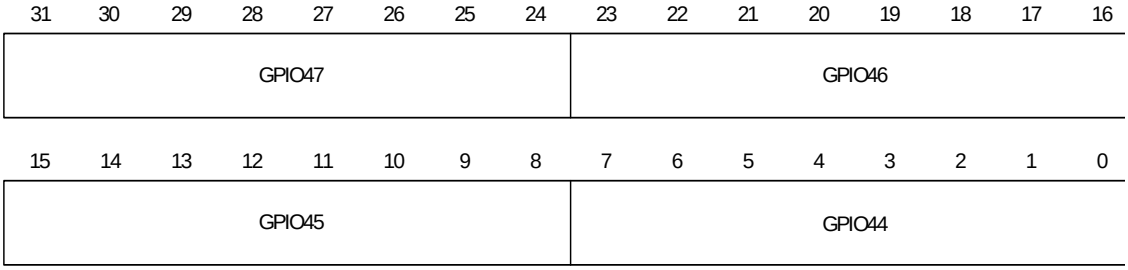
Address: A090_2028



Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO43	0x18	GPIO[43] configuration
D23:16	R/W	GPIO42	0x18	GPIO[42] configuration
D15:08	R/W	GPIO41	0x18	GPIO[41] configuration
D07:00	R/W	GPIO40	0x18	GPIO[40] configuration

GPIO Configuration Register #11

Address: A090_202C

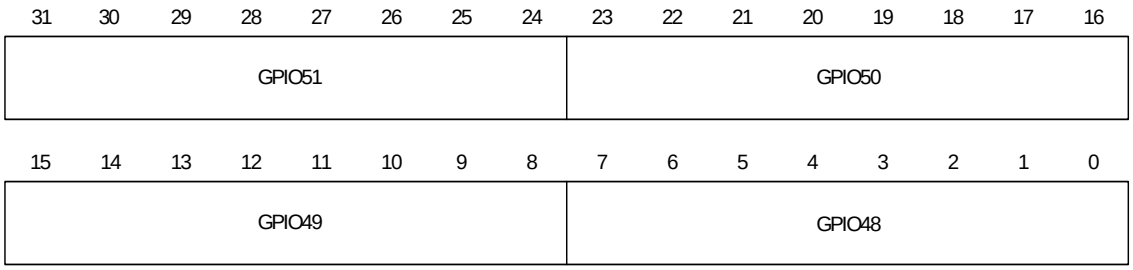


Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO47	0x18	GPIO[47] configuration
D23:16	R/W	GPIO46	0x18	GPIO[46] configuration
D15:08	R/W	GPIO45	0x18	GPIO[45] configuration
D07:00	R/W	GPIO44	0x18	GPIO[44] configuration

Downloaded from www.Manualslib.com manuals search engine

GPIO Configuration Register #12

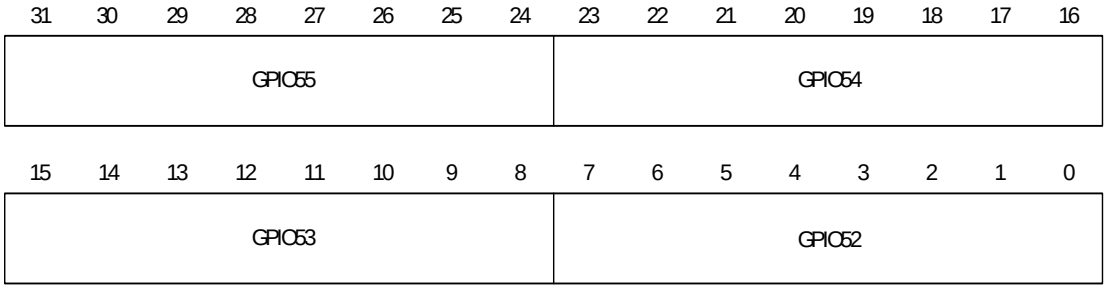
Address: A090_2030



Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO51	0x18	GPIO[51] configuration
D23:16	R/W	GPIO50	0x18	GPIO[50] configuration
D15:08	R/W	GPIO49	0x18	GPIO[49] configuration
D07:00	R/W	GPIO48	0x18	GPIO[48] configuration

GPIO Configuration Register #13

Address: A090_2034

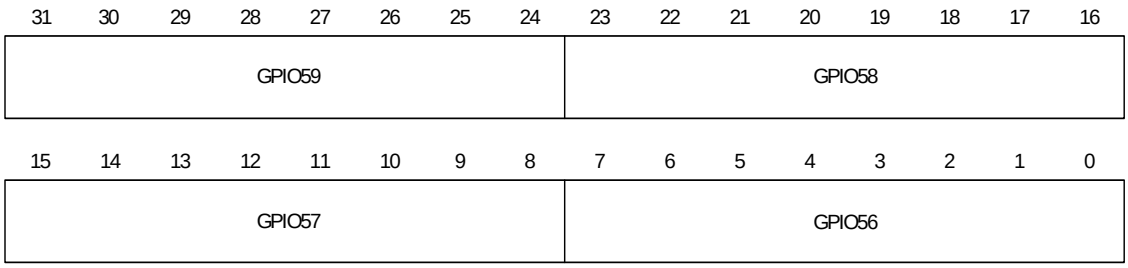


Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO55	0x18	GPIO[55] configuration
D23:16	R/W	GPIO54	0x18	GPIO[54] configuration
D15:08	R/W	GPIO53	0x18	GPIO[53] configuration
D07:00	R/W	GPIO52	0x18	GPIO[52] configuration

Downloaded from www.Manualslib.com manuals search engine

GPIO Configuration Register #14

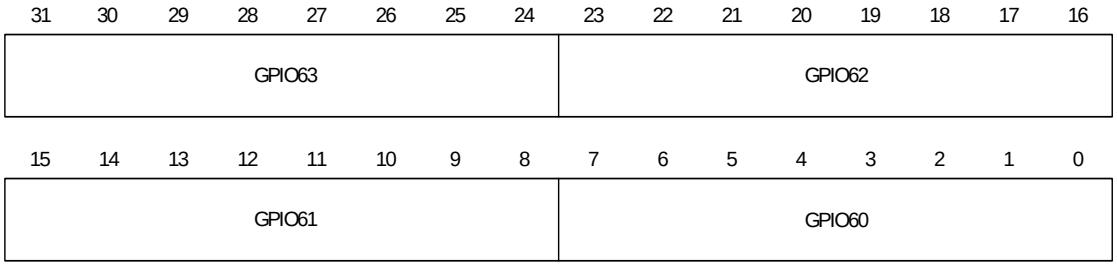
Address: A090_2038



Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO59	0x18	GPIO[59] configuration
D23:16	R/W	GPIO58	0x18	GPIO[58] configuration
D15:08	R/W	GPIO57	0x18	GPIO[57] configuration
D07:00	R/W	GPIO56	0x18	GPIO[56] configuration

GPIO Configuration Register #15

Address: A090_203C

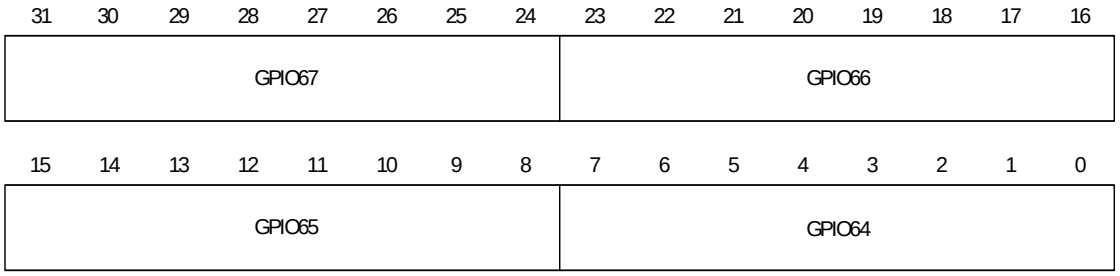


Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO63	0x18	GPIO[63] configuration
D23:16	R/W	GPIO62	0x18	GPIO[62] configuration
D15:08	R/W	GPIO61	0x18	GPIO[61] configuration
D07:00	R/W	GPIO60	0x18	GPIO[60] configuration

Downloaded from www.Manualslib.com manuals search engine

GPIO Configuration Register #16

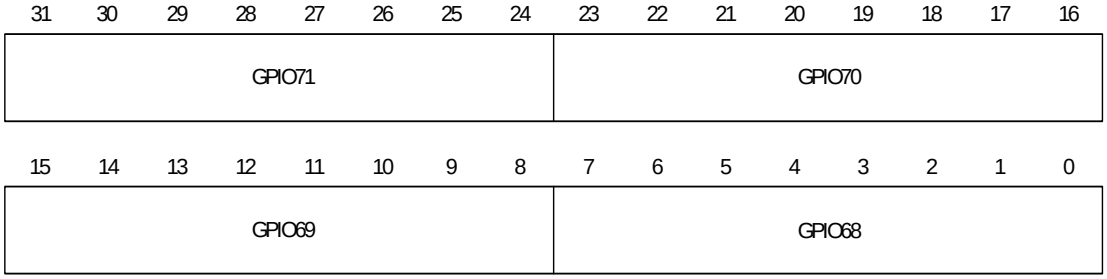
Address: A090_2040



Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO67	0x18	GPIO[67] configuration
D23:16	R/W	GPIO66	0x18	GPIO[66] configuration
D15:08	R/W	GPIO65	0x18	GPIO[65] configuration
D07:00	R/W	GPIO64	0x18	GPIO[64] configuration

GPIO Configuration Register #17

Address: A090_2044

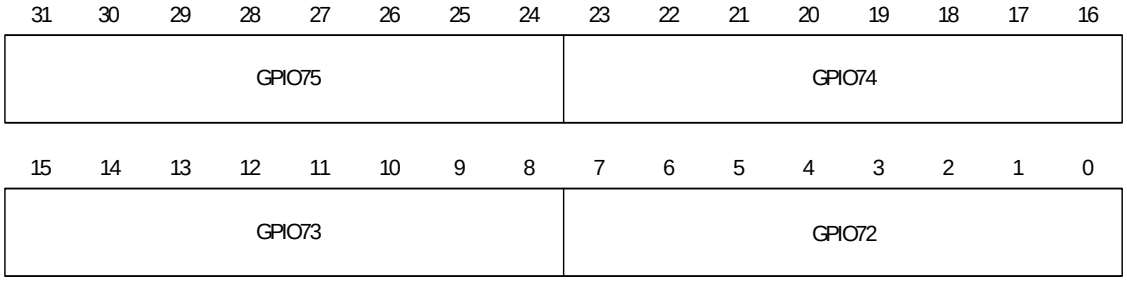


Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO71	0x18	GPIO[71] configuration
D23:16	R/W	GPIO70	0x18	GPIO[70] configuration
D15:08	R/W	GPIO69	0x18	GPIO[69] configuration
D07:00	R/W	GPIO68	0x18	GPIO[68] configuration

Downloaded from www.Manualslib.com manuals search engine

GPIO Configuration Register #18

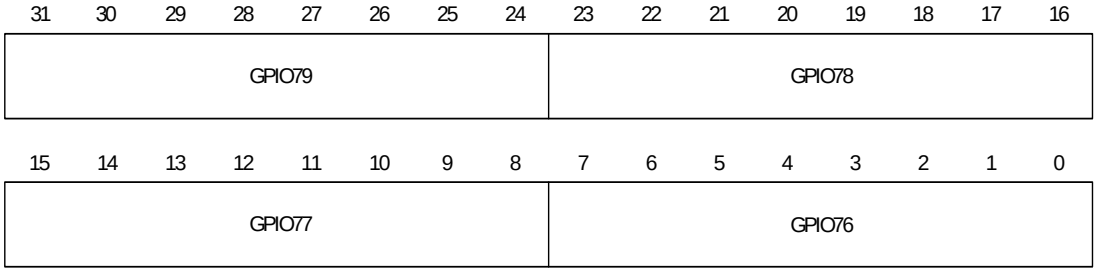
Address: A090_2048



Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO75	0x18	GPIO[75] configuration
D23:16	R/W	GPIO74	0x18	GPIO[74] configuration
D15:08	R/W	GPIO73	0x18	GPIO[73] configuration
D07:00	R/W	GPIO72	0x18	GPIO[72] configuration

GPIO Configuration Register #19

Address: A090_204C

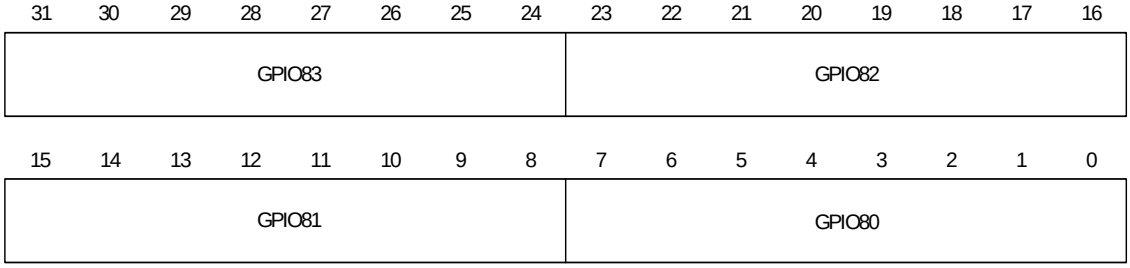


Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO79	0x18	GPIO[79] configuration
D23:16	R/W	GPIO78	0x18	GPIO[78] configuration
D15:08	R/W	GPIO77	0x18	GPIO[77] configuration
D07:00	R/W	GPIO76	0x18	GPIO[76] configuration

Downloaded from www.Manualslib.com manuals search engine

GPIO Configuration Register #20

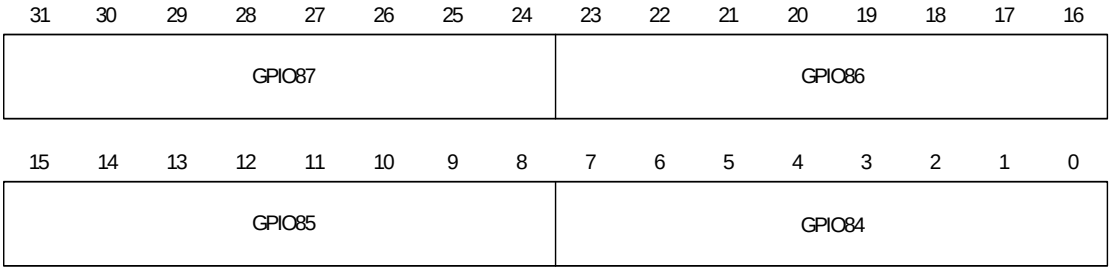
Address: A090_2050



Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO83	0x18	GPIO[83] configuration
D23:16	R/W	GPIO82	0x18	GPIO[82] configuration
D15:08	R/W	GPIO81	0x18	GPIO[81] configuration
D07:00	R/W	GPIO80	0x18	GPIO[80] configuration

GPIO Configuration Register #21

Address: A090_2054

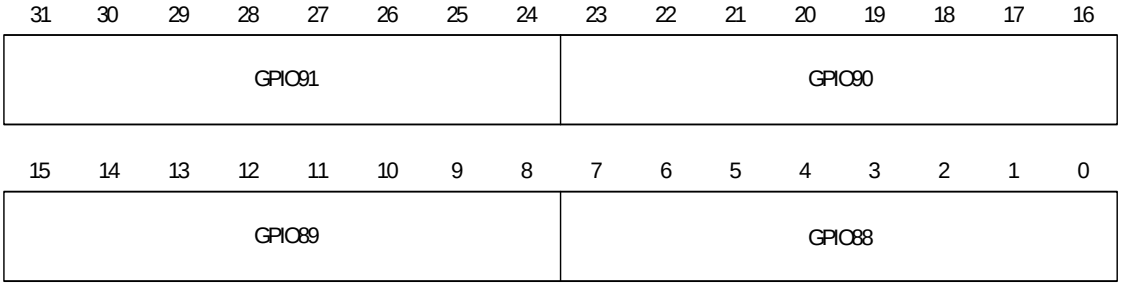


Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO87	0x18	GPIO[87] configuration
D23:16	R/W	GPIO86	0x18	GPIO[86] configuration
D15:08	R/W	GPIO85	0x18	GPIO[85] configuration
D07:00	R/W	GPIO84	0x18	GPIO[84] configuration

Downloaded from www.Manualslib.com manuals search engine

GPIO Configuration Register #22

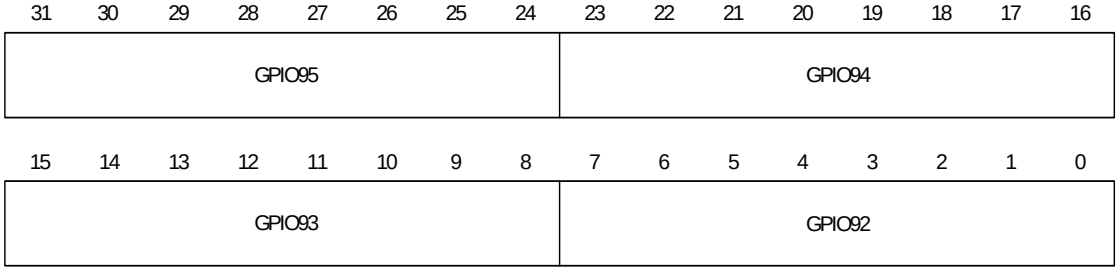
Address: A090_2058



Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO91	0x18	GPIO[91] configuration
D23:16	R/W	GPIO90	0x18	GPIO[90] configuration
D15:08	R/W	GPIO89	0x18	GPIO[89] configuration
D07:00	R/W	GPIO88	0x18	GPIO[88] configuration

GPIO Configuration Register #23

Address: A090_205C

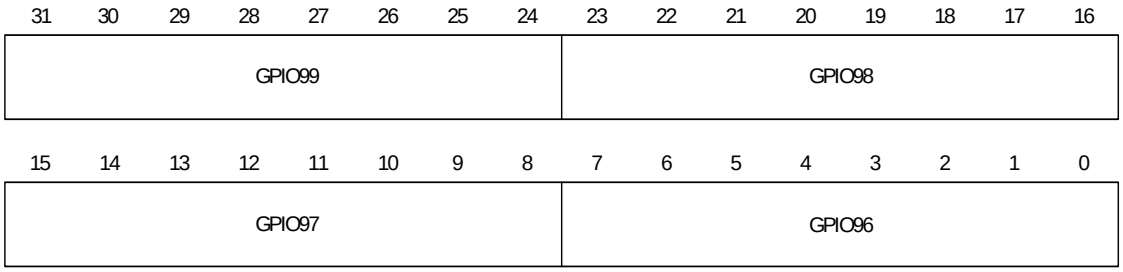


Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO95	0x18	GPIO[95] configuration
D23:16	R/W	GPIO94	0x18	GPIO[94] configuration
D15:08	R/W	GPIO93	0x18	GPIO[93] configuration
D07:00	R/W	GPIO92	0x18	GPIO[92] configuration

Downloaded from www.Manualslib.com manuals search engine

GPIO Configuration Register #24

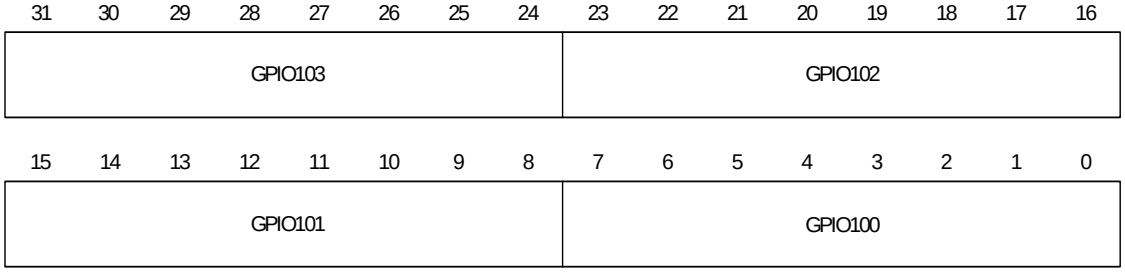
Address: A090_2060



Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO99	0x18	GPIO[99] configuration
D23:16	R/W	GPIO98	0x18	GPIO[98] configuration
D15:08	R/W	GPIO97	0x18	GPIO[97] configuration
D07:00	R/W	GPIO96	0x18	GPIO[96] configuration

GPIO Configuration Register #25

Address: A090_2064

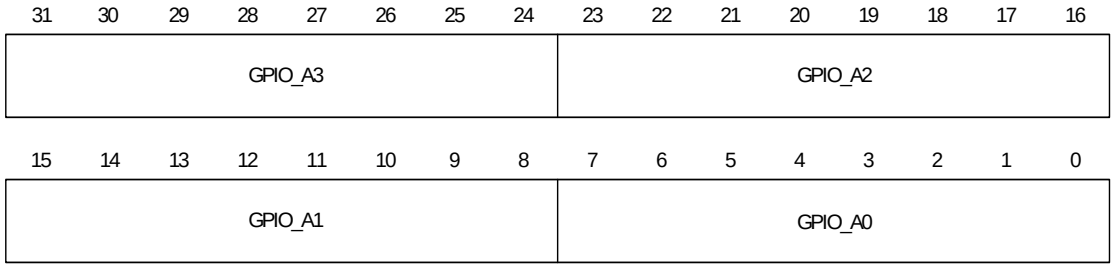


Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO103	0x18	GPIO[103] configuration
D23:16	R/W	GPIO102	0x18	GPIO[102] configuration
D15:08	R/W	GPIO101	0x18	GPIO[101] configuration
D07:00	R/W	GPIO100	0x18	GPIO[100] configuration

Downloaded from www.Manualslib.com manuals search engine

GPIO Configuration Register #26

Address: A090_2068



Bit(s)	Access	Mnemonic	Reset	Description
D31:24	R/W	GPIO_A3	0x18	GPIO_A[3] configuration
D23:16	R/W	GPIO_A2	0x18	GPIO_A[2] configuration
D15:08	R/W	GPIO_A1	0x18	GPIO_A[1] configuration
D07:00	R/W	GPIO_A0	0x18	GPIO_A[0] configuration

Downloaded from www.Manualslib.com manuals search engine

GPIO Control registers

GPIO Control Registers #0 through #3 contain the control information for each of the 108 GPIO pins. When a GPIO pin is configured as a GPIO output, the corresponding bit in GPIO Control Registers #0 through #3 is driven out the GPIO pin. In all configurations, the CPU has read/write access to these registers.

GPIO Control Register #0

Address: A090_206C

Bit(s)	Access	Mnemonic	Reset	Description
D00	R/W	GPIO0	0	GPIO[0] control bit
D01	R/W	GPIO1	0	GPIO[1] control bit
D02	R/W	GPIO2	0	GPIO[2] control bit
D03	R/W	GPIO3	0	GPIO[3] control bit
D04	R/W	GPIO4	0	GPIO[4] control bit
D05	R/W	GPIO5	0	GPIO[5] control bit
D06	R/W	GPIO6	0	GPIO[6] control bit
D07	R/W	GPIO7	0	GPIO[7] control bit
D08	R/W	GPIO8	0	GPIO[8] control bit
D09	R/W	GPIO9	0	GPIO[9] control bit
D10	R/W	GPIO10	0	GPIO[10] control bit
D11	R/W	GPIO11	0	GPIO[11] control bit
D12	R/W	GPIO12	0	GPIO[12] control bit
D13	R/W	GPIO13	0	GPIO[13] control bit
D14	R/W	GPIO14	0	GPIO[14] control bit
D15	R/W	GPIO15	0	GPIO[15] control bit
D16	R/W	GPIO16	0	GPIO[16] control bit
D17	R/W	GPIO17	0	GPIO[17] control bit
D18	R/W	GPIO18	0	GPIO[18] control bit
D19	R/W	GPIO19	0	GPIO[19] control bit
D20	R/W	GPIO20	0	GPIO[20] control bit
D21	R/W	GPIO21	0	GPIO[21] control bit
D22	R/W	GPIO22	0	GPIO[22] control bit
D23	R/W	GPIO23	0	GPIO[23] control bit
D24	R/W	GPIO24	0	GPIO[24] control bit

Downloaded from www.Manualslib.com manuals search engine

Bit(s)	Access	Mnemonic	Reset	Description
D25	R/W	GPIO25	0	GPIO[25] control bit
D26	R/W	GPIO26	0	GPIO[26] control bit
D27	R/W	GPIO27	0	GPIO[27] control bit
D28	R/W	GPIO28	0	GPIO[28] control bit
D29	R/W	GPIO29	0	GPIO[29] control bit
D30	R/W	GPIO30	0	GPIO[30] control bit
D31	R/W	GPIO31	0	GPIO[31] control bit

GPIO Control Register #1

Address: A090_2070

Bit(s)	Access	Mnemonic	Reset	Description
D00	R/W	GPIO32	0	GPIO[32] control bit
D01	R/W	GPIO33	0	GPIO[33] control bit
D02	R/W	GPIO34	0	GPIO[34] control bit
D03	R/W	GPIO35	0	GPIO[35] control bit
D04	R/W	GPIO36	0	GPIO[36] control bit
D05	R/W	GPIO37	0	GPIO[37] control bit
D06	R/W	GPIO38	0	GPIO[38] control bit
D07	R/W	GPIO39	0	GPIO[39] control bit
D08	R/W	GPIO40	0	GPIO[40] control bit
D09	R/W	GPIO41	0	GPIO[41] control bit
D10	R/W	GPIO42	0	GPIO[42] control bit
D11	R/W	GPIO43	0	GPIO[43] control bit
D12	R/W	GPIO44	0	GPIO[44] control bit
D13	R/W	GPIO45	0	GPIO[45] control bit
D14	R/W	GPIO46	0	GPIO[46] control bit
D15	R/W	GPIO47	0	GPIO[47] control bit
D16	R/W	GPIO48	0	GPIO[48] control bit
D17	R/W	GPIO49	0	GPIO[49] control bit
D18	R/W	GPIO50	0	GPIO[50] control bit
D19	R/W	GPIO51	0	GPIO[51] control bit
D20	R/W	GPIO52	0	GPIO[52] control bit
D21	R/W	GPIO53	0	GPIO[53] control bit

Downloaded from www.Manualslib.com manuals search engine

Bit(s)	Access	Mnemonic	Reset	Description
D22	R/W	GPIO54	0	GPIO[54] control bit
D23	R/W	GPIO55	0	GPIO[55] control bit
D24	R/W	GPIO56	0	GPIO[56] control bit
D25	R/W	GPIO57	0	GPIO[57] control bit
D26	R/W	GPIO58	0	GPIO[58] control bit
D27	R/W	GPIO59	0	GPIO[59] control bit
D28	R/W	GPIO60	0	GPIO[60] control bit
D29	R/W	GPIO61	0	GPIO[61] control bit
D30	R/W	GPIO62	0	GPIO[62] control bit
D31	R/W	GPIO63	0	GPIO[63] control bit

GPIO Control Register #2

Address: A090_2074

Bit(s)	Access	Mnemonic	Reset	Description
D00	R/W	GPIO64	0	GPIO[64] control bit
D01	R/W	GPIO65	0	GPIO[65] control bit
D02	R/W	GPIO66	0	GPIO[66] control bit
D03	R/W	GPIO67	0	GPIO[67] control bit
D04	R/W	GPIO68	0	GPIO[68] control bit
D05	R/W	GPIO69	0	GPIO[69] control bit
D06	R/W	GPIO70	0	GPIO[70] control bit
D07	R/W	GPIO71	0	GPIO[71] control bit
D08	R/W	GPIO72	0	GPIO[72] control bit
D09	R/W	GPIO73	0	GPIO[73] control bit
D10	R/W	GPIO74	0	GPIO[74] control bit
D11	R/W	GPIO75	0	GPIO[75] control bit
D12	R/W	GPIO76	0	GPIO[76] control bit
D13	R/W	GPIO77	0	GPIO[77] control bit
D14	R/W	GPIO78	0	GPIO[78] control bit
D15	R/W	GPIO79	0	GPIO[79] control bit
D16	R/W	GPIO80	0	GPIO[80] control bit
D17	R/W	GPIO81	0	GPIO[81] control bit
D18	R/W	GPIO82	0	GPIO[82] control bit

Downloaded from www.Manualslib.com manuals search engine

Bit(s)	Access	Mnemonic	Reset	Description
D19	R/W	GPIO83	0	GPIO[83] control bit
D20	R/W	GPIO84	0	GPIO[84] control bit
D21	R/W	GPIO85	0	GPIO[85] control bit
D22	R/W	GPIO86	0	GPIO[86] control bit
D23	R/W	GPIO87	0	GPIO[87] control bit
D24	R/W	GPIO88	0	GPIO[88] control bit
D25	R/W	GPIO89	0	GPIO[89] control bit
D26	R/W	GPIO90	0	GPIO[90] control bit
D27	R/W	GPIO91	0	GPIO[91] control bit
D28	R/W	GPIO92	0	GPIO[92] control bit
D29	R/W	GPIO93	0	GPIO[93] control bit
D30	R/W	GPIO94	0	GPIO[94] control bit
D31	R/W	GPIO95	0	GPIO[95] control bit

GPIO Control Register #3

Address: A090_2078

Bit(s)	Access	Mnemonic	Reset	Description
D00	R/W	GPIO96	0	GPIO[96] control bit
D01	R/W	GPIO97	0	GPIO[97] control bit
D02	R/W	GPIO98	0	GPIO[98] control bit
D03	R/W	GPIO99	0	GPIO[99] control bit
D04	R/W	GPIO100	0	GPIO[100] control bit
D05	R/W	GPIO101	0	GPIO[101] control bit
D06	R/W	GPIO102	0	GPIO[102] control bit
D07	R/W	GPIO103	0	GPIO[103] control bit
D08	R/W	GPIO_A0	0	GPIO_A[0] control bit
D09	R/W	GPIO_A1	0	GPIO_A[1] control bit
D10	R/W	GPIO_A2	0	GPIO_A[2] control bit
D11	R/W	GPIO_A3	0	GPIO_A[3] control bit
D31:12	N/A	Reserved	N/A	N/A

Downloaded from www.Manualslib.com manuals search engine

GPIO Status registers

GPIO Status Registers #0 through #3 contain the status information for each of the 108 GPIO pins. In all configurations, the value on the GPIO input pin is brought to the status register and the CPU has read-only access to the register.

GPIO Status Register #1

Address: A090_2080

Bit(s)	Access	Mnemonic	Reset	Description
D00	R	GPIO32	Undefined	GPIO[32] status bit
D01	R	GPIO33	Undefined	GPIO[33] status bit
D02	R	GPIO34	Undefined	GPIO[34] status bit
D03	R	GPIO35	Undefined	GPIO[35] status bit
D04	R	GPIO36	Undefined	GPIO[36] status bit
D05	R	GPIO37	Undefined	GPIO[37] status bit
D06	R	GPIO38	Undefined	GPIO[38] status bit
D07	R	GPIO39	Undefined	GPIO[39] status bit
D08	R	GPIO40	Undefined	GPIO[40] status bit
D09	R	GPIO41	Undefined	GPIO[41] status bit
D10	R	GPIO42	Undefined	GPIO[42] status bit
D11	R	GPIO43	Undefined	GPIO[43] status bit
D12	R	GPIO44	Undefined	GPIO[44] status bit
D13	R	GPIO45	Undefined	GPIO[45] status bit
D14	R	GPIO46	Undefined	GPIO[46] status bit
D15	R	GPIO47	Undefined	GPIO[47] status bit
D16	R	GPIO48	Undefined	GPIO[48] status bit
D17	R	GPIO49	Undefined	GPIO[49] status bit
D18	R	GPIO50	Undefined	GPIO[50] status bit
D19	R	GPIO51	Undefined	GPIO[51] status bit
D20	R	GPIO52	Undefined	GPIO[52] status bit
D21	R	GPIO53	Undefined	GPIO[3] status bit
D22	R	GPIO54	Undefined	GPIO[54] status bit
D23	R	GPIO55	Undefined	GPIO[55] status bit
D24	R	GPIO56	Undefined	GPIO[56] status bit
D25	R	GPIO57	Undefined	GPIO[57] status bit

Downloaded from www.Manualslib.com manuals search engine

Bit(s)	Access	Mnemonic	Reset	Description
D26	R	GPIO58	Undefined	GPIO[58] status bit
D27	R	GPIO59	Undefined	GPIO[59] status bit
D28	R	GPIO60	Undefined	GPIO[60] status bit
D29	R	GPIO61	Undefined	GPIO[61] status bit
D30	R	GPIO62	Undefined	GPIO[62] status bit
D31	R	GPIO63	Undefined	GPIO[63] status bit

GPIO Status Register #2

Address: A090_2084

Bit(s)	Access	Mnemonic	Reset	Description
D00	R	GPIO64	Undefined	GPIO[64] status bit
D01	R	GPIO65	Undefined	GPIO[65] status bit
D02	R	GPIO66	Undefined	GPIO[66] status bit
D03	R	GPIO67	Undefined	GPIO[67] status bit
D04	R	GPIO68	Undefined	GPIO[68] status bit
D05	R	GPIO69	Undefined	GPIO[69] status bit
D06	R	GPIO70	Undefined	GPIO[70] status bit
D07	R	GPIO71	Undefined	GPIO[71] status bit
D08	R	GPIO72	Undefined	GPIO[72] status bit
D09	R	GPIO73	Undefined	GPIO[73] status bit
D10	R	GPIO74	Undefined	GPIO[74] status bit
D11	R	GPIO75	Undefined	GPIO[75] status bit
D12	R	GPIO76	Undefined	GPIO[76] status bit
D13	R	GPIO77	Undefined	GPIO[77] status bit
D14	R	GPIO78	Undefined	GPIO[78] status bit
D15	R	GPIO79	Undefined	GPIO[79] status bit
D16	R	GPIO80	Undefined	GPIO[80] status bit
D17	R	GPIO81	Undefined	GPIO[81] status bit
D18	R	GPIO82	Undefined	GPIO[82] status bit
D19	R	GPIO83	Undefined	GPIO[83] status bit
D20	R	GPIO84	Undefined	GPIO[84] status bit
D21	R	GPIO85	Undefined	GPIO[85] status bit
D22	R	GPIO86	Undefined	GPIO[86] status bit

Downloaded from www.Manualslib.com manuals search engine

Bit(s)	Access	Mnemonic	Reset	Description
D23	R	GPIO87	Undefined	GPIO[87] status bit
D24	R	GPIO88	Undefined	GPIO[88] status bit
D25	R	GPIO89	Undefined	GPIO[89] status bit
D26	R	GPIO90	Undefined	GPIO[90] status bit
D27	R	GPIO91	Undefined	GPIO[91] status bit
D28	R	GPIO92	Undefined	GPIO[92] status bit
D29	R	GPIO93	Undefined	GPIO[93] status bit
D30	R	GPIO94	Undefined	GPIO[94] status bit
D31	R	GPIO95	Undefined	GPIO[95] status bit

GPIO Status Register #3

Address: A090_2088

Bit(s)	Access	Mnemonic	Reset	Description
D00	R	GPIO96	Undefined	GPIO[96] status bit
D01	R	GPIO97	Undefined	GPIO[97] status bit
D02	R	GPIO98	Undefined	GPIO[98] status bit
D03	R	GPIO99	Undefined	GPIO[99] status bit
D04	R	GPIO100	Undefined	GPIO[100] status bit
D05	R	GPIO101	Undefined	GPIO[101] status bit
D06	R	GPIO102	Undefined	GPIO[102] status bit
D07	R	GPIO103	Undefined	GPIO[103] status bit
D08	R	GPIO_A0	Undefined	GPIO_A[0] status bit
D09	R	GPIO_A1	Undefined	GPIO_A[1] status bit
D10	R	GPIO_A2	Undefined	GPIO_A[2] status bit
D11	R	GPIO_A3	Undefined	GPIO_A[3] status bit
D31:12	N/A	Reserved	N/A	N/A

Memory Bus Configuration register

The Memory Bus Configuration register controls chip select and upper address options.

Address: A090_208C

Downloaded from www.Manualslib.com manuals search engine

Bit(s)	Access	Mnemonic	Reset	Description
D02:00	R/W	CS0	0x4	Controls which system memory chip select is routed to CS0 000 dy_cs_0 001 dy_cs_1 010 dy_cs_2 011 dy_cs_3 100 st_cs_0 (default) 101 st_cs_1 110 st_cs_2 111 st_cs_3
D05:03	R/W	CS1	0x0	Controls which system memory chip select is routed to CS1 000 dy_cs_0 (default) 001 dy_cs_1 010 dy_cs_2 011 dy_cs_3 100 st_cs_0 101 st_cs_1 110 st_cs_2 111 st_cs_3
D08:06	R/W	CS2	0x5	Controls which system memory chip select is routed to CS2 000 dy_cs_0 001 dy_cs_1 010 dy_cs_2 011 dy_cs_3 100 st_cs_0 101 st_cs_1 (default) 110 st_cs_2 111 st_cs_3
D11:09	R/W	CS3	0x1	Controls which system memory chip select is routed to CS3 000 dy_cs_0 001 dy_cs_1 (default) 010 dy_cs_2 011 dy_cs_3 100 st_cs_0 101 st_cs_1 110 st_cs_2 111 st_cs_3

Downloaded from www.Manualslib.com manuals search engine

Bit(s)	Access	Mnemonic	Reset	Description
D14:12	R/W	CS4	0x6	Controls which system memory chip select is routed to CS4 000 dy_cs_0 001 dy_cs_1 010 dy_cs_2 011 dy_cs_3 100 st_cs_0 101 st_cs_1 110 st_cs_2 (default) 111 st_cs_3
D17:15	R/W	CS5	0x2	Controls which system memory chip select is routed to CS5 000 dy_cs_0 001 dy_cs_1 010 dy_cs_2 (default) 011 dy_cs_3 100 st_cs_0 101 st_cs_1 110 st_cs_2 111 st_cs_3
D20:18	R/W	CS6	0x7	Controls which system memory chip select is routed to CS6 000 dy_cs_0 001 dy_cs_1 010 dy_cs_2 011 dy_cs_3 100 st_cs_0 101 st_cs_1 110 st_cs_2 111 st_cs_3 (default)
D23:21	R/W	CS7	0x3	Controls which system memory chip select is routed to CS7 000 dy_cs_0 001 dy_cs_1 010 dy_cs_2 011 dy_cs_3 (default) 100 st_cs_0 101 st_cs_1 110 st_cs_2 111 st_cs_3
D24	R/W	DHPUDIS	0x0	**High data bus pullup control** 0 Enable pullup resistors on data[31:16] 1 Disable pullup resistors on data[31:16] Note: Bits 15:00 are output and controlled through GPIO

Downloaded from www.Manualslib.com manuals search engine

Bit(s)	Access	Mnemonic	Reset	Description
D25	R/W	APUDIS	0x0	**Address bus pullup control** **(Applicable only to address associated with hardware strapping)** 0 Enable pullup resistors 1 Disable pullup resistors Note: Bits 27:24 are output and controlled through GPIO
D31:26	N/A	Reserved	N/A	N/A

Downloaded from www.Manualslib.com manuals search engine

Downloaded from www.Manualslib.com manuals search engine

Working with the CPU

CHAPTER 3

This processor core is based on the ARM926EJ-S processor. The ARM926EJ-S processor belongs to the ARM9 family of general-purpose microprocessors. The ARM926EJ-S processor is targeted at multi-tasking applications in which full memory management, high performance, low die size, and low power are important.

About the processor

The ARM926EJ-S processor supports the 32-bit ARM and 16-bit Thumb instructions sets, allowing you to trade off between high performance and high code density. The processor includes features for efficient execution of Java byte codes, providing Java performance similar to JIT but without the associated overhead.

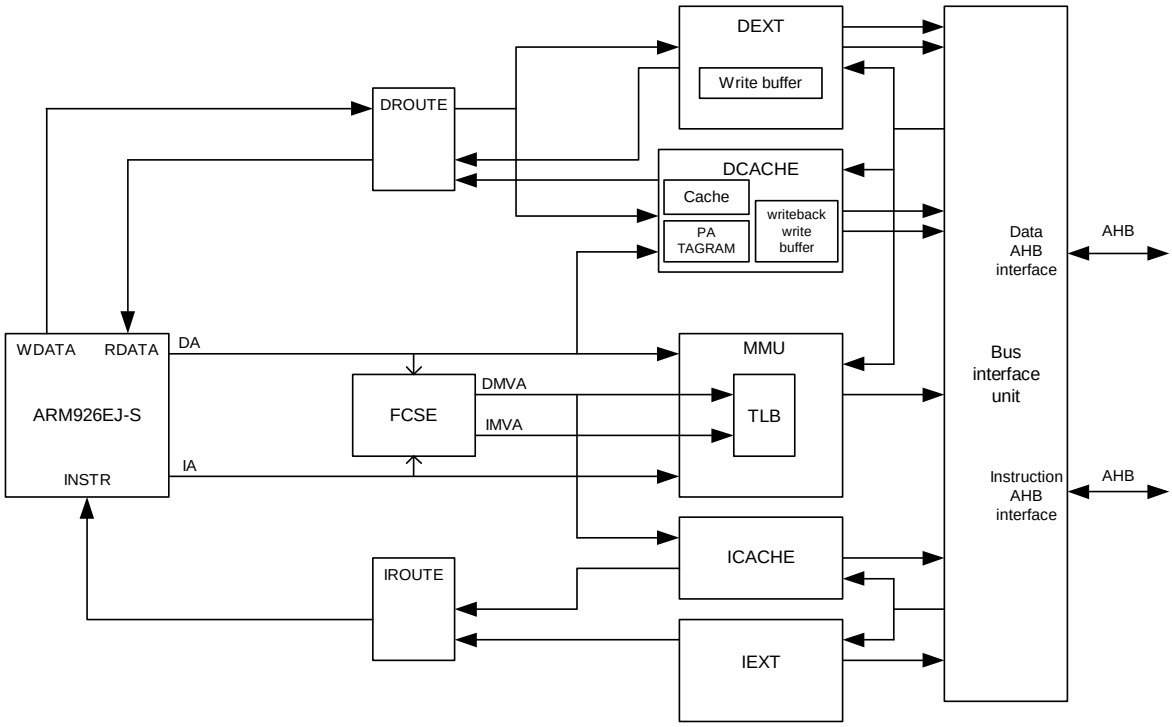
The ARM926EJ-S supports the ARM debug architecture, and includes logic to assist in both hardware and software debug. The processor has a Harvard-cached architecture and provides a complete high-performance processor subsystem, including:

- ARM926EJ-S integer core
- Memory Management Unit (MMU) (see "MemoryManagement Unit (MMU)," beginning on page 105, for information)
- Separate instruction and data AMBA AHB bus interfaces

Downloaded from www.Manualslib.com manuals search engine

Arm926EJ-S process block diagram

This drawing shows the main blocks in the ARM926EJ-S processor.



Instruction sets

The processor executes three instruction sets:

- 32-bit ARM instruction set
- 16-bit Thumb instruction set
- 8-bit Java instruction set

ARM instruction set

The ARM instruction set allows a program to achieve maximum performance with the minimum number of instructions. The majority of instructions are executed in a single cycle.

Thumb instruction set

The Thumb instruction set is simpler than the ARM instruction set, and offers increased code density for code that does not require maximum performance. Code can switch between ARM and Thumb instruction sets on any procedure call.

Downloaded from www.Manualslib.com manuals search engine

Java instruction set

In Java state, the processor core executes a majority of Java bytecodes naturally. Bytecodes are decoded in two states, compared to a single decode stage when in ARM/Thumb mode. See “Jazelle(Java)” on page 104 for more information about Java.

System control processor (CP15) registers

The system control processor (CP15) registers configure and control most of the options in the ARM926EJ-S processor. Access the CP15 registers using only the MRC and MCR instructions in a privileged mode; the instructions are provided in the explanation of each applicable register. Using other instructions, or MRC and MCR in unprivileged mode, results in an UNDEFINED instruction exception.

ARM926EJ-S system addresses

The ARM926EJ-S has three distinct types of addresses:

- In the ARM926EJ-S domain: Virtual address (VA)
- In the Cache and MMU domain: Modified virtual address (MVA)
- In the AMBA domain: Physical address (PA)

Address manipulation example

This is an example of the address manipulation that occurs when the ARM926EJ-S core requests an instruction:

1. The ARM926EJ-S core issues the virtual address of the instruction.
2. The virtual address is translated using the FCSE PID (fast context switch extension process ID) value to the modified virtual address. The instruction cache (ICache) and memory management unit (MMU) find the modified virtual address (see “R13:Process ID register” on page 102).
3. If the protection check carried out by the MMU on the modified virtual address does not abort and the modified virtual address tag is in the ICache, the instruction data is returned to the ARM926EJ-S core.

   If the protection check carried out by the MMU on the modified virtual address does not abort but the cache misses (the MVA tag is not in the cache), the MMU translates the modified virtual address to produce the physical address. This address is given to the AMBA bus interface to perform an external access.

Accessing CP15 registers

Use only MRC and MCR instructions, only in privileged mode, to access CP15 registers. Figure 1 shows the MRC and MCR instruction bit pattern.

Downloaded from www.Manualslib.com manuals search engine

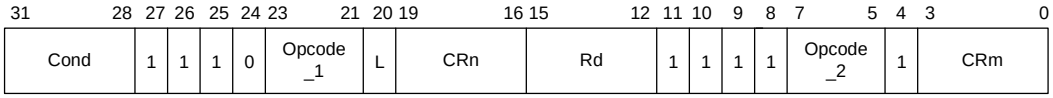


Figure 1: CP15 MRC and MCR bit pattern

The mnemonics for these instructions are:

MCR{cond} p15,opcode_1,Rd,CRn,CRm,opcode_2

MRC{cond} p15,opcode_1,Rd,CRn,CRm,opcode_2

If you try to read from a write-only register or write to a read-only register, you will have UNPREDICTABLE results. In all instructions that access CP15:

- The opcode_1 field SHOULD BE ZERO, except when the values specified are used to select the operations you want. Using other values results in unpredictable behavior.
- The opcode_2 and CRm fields SHOULD BE ZERO, except when the values specified are used to select the behavior you want. Using other values results in unpredictable behavior.

Terms and abbreviations

This table lists the terms and abbreviations used in the CP15 registers and explanations.

Term	Abbreviation	Description
UNPREDICTABLE	UNP	**For reads:** The data returned when reading from this location is unpredictable, and can have any value. **For writes:** Writing to this location causes unpredictable behavior, or an unpredictable change in device configuration.
UNDEFINED	UND	An instruction that accesses CP15 in the manner indicated takes the UNDEFINED instruction exception.
SHOULD BE ZERO	SBZ	When writing to this field, all bits of the field SHOULD BE ZERO.
SHOULD BE ONE	SBO	When writing to this location, all bits in this field SHOULD BE ONE.
SHOULD BE ZERO or PRESERVED	SBZP	When writing to this location, all bits of this field SHOULD BE ZERO or PRESERVED by writing the same value that has been read previously from the same field.

Downloaded from www.Manualslib.com manuals search engine

Note: In all cases, reading from or writing any data values to any CP15 registers, including those fields specified as UNPREDICTABLE, SHOULD BE ONE, or SHOULD BE ZERO, does not cause any physical damage to the chip.

Register summary

CP15 uses 16 registers.

- Register locations 0, 5, and 13 each provide access to more than one register. The register accessed depends on the value of the opcode_2 field in the CP15 MRC/MCR instructions (see "Accessing CP15 registers" on page 83).
- Register location 9 provides access to more than one register. The register accessed depends on the value of the CRm field (see "Accessing CP15 registers" on page 83).

Register	Reads	Writes
0	ID code (based on opcode_2 value)	Unpredictable
0	Cache type (based on opcode_2 value)	Unpredictable
1	Control	Control
2	Translation table base	Translation table base
3	Domain access control	Domain access control
4	Reserved	Reserved
5	Data fault status (based on opcode_2 value)	Data fault status (based on opcode_2 value)
6	Instruction fault status (based on opcode_2 value)	Instruction fault status (based on opcode_2 value)
7	Cache operations	Cache operations
8	Unpredictable	TLB
9	Cache lockdown (based on CRm value)	Cache lockdown
10	TLB lockdown	TLB lockdown
11 and 12	Reserved	Reserved
13	FCSE PID (based on opcode_2 value) FCSE = Fast context switch extension PID = Process identifier	FCSE PID (based on opcode_2 value) FCSE = Fast context switch extension PID = Process identifier
13	Context ID (based on opcode_2 value)	Context ID (based on opcode_2 value)
14	Reserved	Reserved
15	Test configuration	Test configuration

All CP15 register bits that are defined and contain state are set to 0 by reset, with these exceptions:

- The V bit is set to 0 at reset if the VINITHI signal is low, and set to 1 if the VINITHI signal is high.

Downloaded from www.Manualslib.com manuals search engine

- The B bit is set to 0 at reset if the BIGENDINIT signal is low, and set to 1 if the BIGENDINIT signal is high.

R0: ID code and cache type status registers

Register R0 access the ID register, and cache type register. Reading from R0 returns the device ID, and the cache type, depending on the opcode_2 value:

opcode_2=0	ID value
opcode_2=1	instruction and data cache type

The CRm field SHOULD BE ZERO when reading from these registers. This table shows the instructions you can use to read register R0.

Function	Instruction
Read ID code	MRC p15,0,Rd,c0,c0,{0, 3-7}
Read cache type	MRC p15,0,Rd,c0,c0,1

Writing to register R0 is UNPREDICTABLE.

R0: ID code

R0: ID code is a read-only register that returns the 32-bit device ID code. You can access the ID code register by reading CP15 register R0 with the opcode_2 field set to any value other than 1 or 2. Note this example:

```
MRC p15, 0, Rd, c0, c0, {0, 3-7}; returns ID
```

This is the contents of the ID code register.

Bits	Function	Value
[31:24]	ASCII code of implementer trademark	0x41
[23:20]	Specification revision	0x0
[19:16]	Architecture (ARMv5TEJ)	0x6
[15:4]	Part number	0x926
[3:0]	Layout revision	0x0

R0: Cache type register

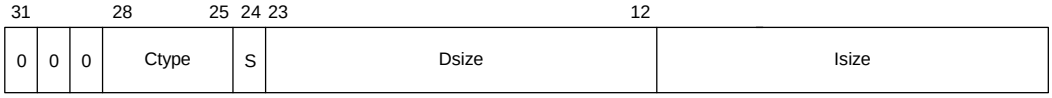
R0: Cache type is a read-only register that contains information about the size and architecture of the instruction cache (ICache) and data cache (DCache) enabling operating systems to establish how to perform operations such as cache cleaning and lockdown. See “Cache features” on page 127 for more information about cache.

Downloaded from www.Manualslib.com manuals search engine

You can access the cache type register by reading CP15 register R0 with the opcode_2 field set to 1. Note this example:

```
MRC p15, 0, Rd, c0, c0, 1; returns cache details
```

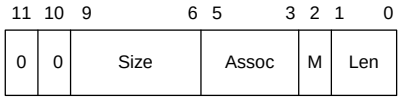
Cache type register and field description



Field	Description
Ctype	Determines the cache type, and specifies whether the cache supports lockdown and how it is cleaned. Ctype encoding is shown below; all unused values are reserved. Value: 0b1110 Method: Writeback Cache cleaning: Register 7 operations (see “R7:Cache Operations register” on page 94) Cache lockdown: Format C (see “R9: Cache Lockdown register” on page 98)
S bit	Specifies whether the cache is a unified cache (S=0) or separate ICache and DCache (S=1). Will always report separate ICache and DCache for this processor.
Dsize	Specifies the size, line length, and associativity of the DCache.
Isize	Species the size, length and associativity of the ICache.

Dsize and Isize fields

The Dsize and Isize fields in the cache type register have the same format, as shown:



The field contains these bits:

Downloaded from www.Manualslib.com manuals search engine

Field	Description
Size	Determines the cache size in conjunction with the M bit. ■ The M bit is 0 for DCache and ICache. ■ The size field is bits [21:18] for the DCache and bits [9:6] for the ICache. ■ The minimum size of each cache is 4 KB; the maximum size is 128 KB. ■ Cache size encoding with M=0: **Size field** — **Cache size** 0b0011 — 4 KB 0b0100 — 8 KB Note: The processor always reports 4KB for DCache and 8KB for ICache.
Assoc	Determines the cache associativity in conjunction with the M bit. ■ The M bit is 0 for both DCache and ICache. ■ The assoc field is bits [17:15 for the DCache and bits [5:3] for the ICache. ■ Cache associativity with encoding: **Assoc field** — **Associativity** 0b010 — 4-way Other values — Reserved
M bit	Multiplier bit. Determines the cache size and cache associativity values in conjunction with the size and assoc fields. Note: This field must be set to 0 for the ARM926EJ-S processor.
Len	Determines the line length of the cache. ■ The len field is bits [13:12] for the DCache and bits [1:0] for the ICache. ■ Line length encoding: **Len field** — **Cache line length** 10 — 8 words (32 bytes) Other values — Reserved

R1: Control register

Register R1 is the control register for the ARM926EJ-S processor. This register specifies the configuration used to enable and disable the caches and MMU (memory management unit). It is recommended that you access this register using a read-modify-write sequence.

For both reading and writing, the CRm and opcode_2 fields SHOULD BE ZERO. Use these instructions to read and write this register:

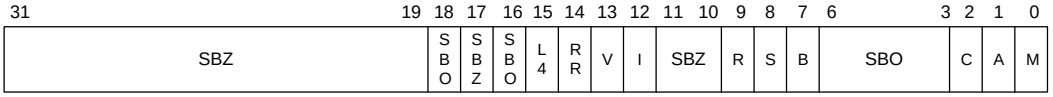
```
MRC p15, 0, Rd, c1, c0, 0; read control register
MCR p15, Rd, c1, c0, 0; write control register
```

All defined control bits are set to zero on reset except the V bit and B bit.

- The V bit is set to zero at reset if the VINITHI signal is low.
- The B bit is set to zero at reset if the BIGENDINIT signal is low, and set to one if the BIGENDINIT signal is high.

Downloaded from www.Manualslib.com manuals search engine

Control register



Bit functionality

Bits	Name	Function
[31:19]	N/A	Reserved: ■ When read, returns an UNPREDICTABLE value. ■ When written, SHOULD BE ZERO, or a value read from bits [31:19] on the same processor. ■ Use a read-modify-write sequence when modifying this register to provide the greatest future compatibility.
[18]	N/A	Reserved, SBO. Read = 1, write =1.
[17]	N/A	Reserved, SBZ. read = 0, write = 0.
[16]	N/A	Reserved, SBO. Read = 1, write = 1.
[15]	L4	Determines whether the T is set when load instructions change the PC. 0 Loads to PC set the T bit 1 Loads to PC do not set the T bit
[14]	RR bit	**Replacement strategy for ICache and DCache** 0 Random replacement 1 Round-robin replacement
[13]	V bit	**Location of exception vectors** 0 Normal exception vectors selected; address range=0x0000 0000 to 0x0000 001C 1 High exception vectors selected; address range=0xFFFF 0000 to 0xFFFF 001C Set to the value of VINITHI on reset.
[12]	I bit	**ICache enable/disable** 0 ICache disabled 1 ICache enabled
[11:10]	N/A	SHOULD BE ZERO
[9]	R bit	**ROM protection** Modifies the ROM protection system.
[8]	S bit	**System protection** Modifies the MMU protection system. See "MemoryManagement Unit (MMU)," beginning on page 105.
[7]	B bit	Endianness 0 Little endian operation 1 Big endian operation Set to the value of BIGENDINIT on reset.

Downloaded from www.Manualslib.com manuals search engine

Bits	Name	Function
[6:3]	N/A	Reserved. SHOULD BE ONE.
[2]	C bit	**DCache enable/disable** 0 Cache disabled 1 Cache enabled
[1]	A bit	**Alignment fault enable/disable** 0 Data address alignment fault checking disabled 1 Data address alignment fault checking enabled
[0]	M bit	**MMU enable/disable** 0 Disabled 1 Enabled

ICache and DCache behavior

The M, C, I, and RR bits directly affect ICache and DCache behavior, as shown:

Cache	MMU	Behavior
ICache disabled	Enabled or disabled	All instruction fetches are from external memory (AHB).
ICache enabled	Disabled	All instruction fetches are cachable, with no protection checking. All addresses are flat-mapped; that is: VA=MVA=PA.
ICache enabled	Enabled	Instruction fetches are cachable or noncachable, and protection checks are performed. All addresses are remapped from VA to PA, depending on the MMU page table entry; that is, VA translated to MVA, MVA remapped to PA.
DCache disabled	Enabled or disabled	All data accesses are to external memory (AHB).
DCache enabled	Disabled	All data accesses are noncachable nonbufferable. All addresses are flat-mapped; that is, VA=MVA=PA.
DCache enabled	Enabled	All data accesses are cachable or noncachable, and protection checks are performed. All addresses are remapped from VA to PA, depending on the MMU page table entry; that is, VA translated to MVA, MVA remapped to PA.

If either the DCache or ICache is disabled, the contents of that cache are not accessed. If the cache subsequently is re-enabled, the contents will not have changed. To guarantee that memory coherency is maintained, the DCache must be cleaned of dirty data before it is disabled.

Downloaded from www.Manualslib.com manuals search engine

R2: Translation Table Base register

Register R2 is the Translation Table Base register (TTBR), for the base address of the first-level translation table.

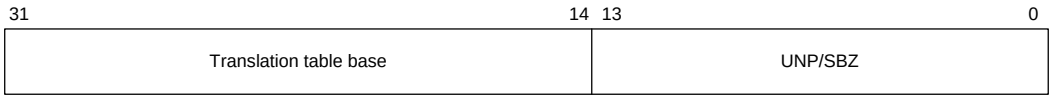
- Reading from R2 returns the pointer to the currently active first-level translation table in bits [31:14] and an UNPREDICTABLE value in bits [13:0].
- Writing to R2 updates the pointer to the first-level translation table from the value in bits[31:14] of the written value. Bits [13:0] SHOULD BE ZERO.

Use these instructions to access the Translation Table Base register:

```
MRC p15, 0, Rd, c2, c0, 0; read TTBR
MCR p15, 0, Rd, c2, c0, 0; write TTBR
```

The CRm and opcode_2 fields SHOULD BE ZERO when writing to R2.

Register format

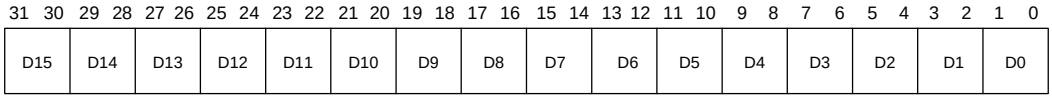


R3:Domain Access Control register

Register R3 is the Domain Access Control register and consists of 16 two-bit fields.

- Reading from R3 returns the value of the Domain Access Control register.
- Writing to R3 writes the value of the Domain Access Control register.

Register format



Access permissions and instructions

Each two-bit field defines the access permissions for one of the 16 domains (D15–D0):

00	No access: Any access generates a domain fault
01	Client: Accesses are checked against the access permission bits in the section or page descriptor
10	Reserved: Currently behaves like no access mode (00)
11	Manager: Accesses are not checked against the access permission bits, so a permission fault cannot be generated.

Use these instructions to access the Domain Access Control register:

```
MRC p15, 0, Rd, c3, c0, 0; read domain access permissions
MCR p15, 0, Rd, c3, c0, 0; write domain access permissions
```


Downloaded from www.Manualslib.com manuals search engine

R4 register

Accessing (reading or writing) this register causes UNPREDICTABLE behavior.

R5: Fault Status registers

Register R5 accesses the Fault Status registers (FSRs). The Fault Status registers contain the source of the last instruction or data fault. The instruction-side FSR is intended for debug purposes only.

The FSR is updated for alignment faults and for external aborts that occur while the MMU is disabled. The FSR accessed is determined by the opcode_2 value:

opcode_2=0	Data Fault Status register (DFSR)
opcode_2=1	Instruction Fault Status register (IFSR)

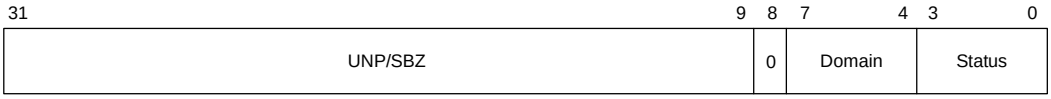
See "MemoryManagement Unit (MMU)," beginning on page 105, for the fault type encoding.

Access instructions

Access the FSRs using these instructions:

```
MRC p15, 0, Rd, c5, c0, 0; read DFSR
MCR p15, 0, Rd, c5, c0, 0; write DFSR
MRC p15, 0, Rd, c5, c0, 1; read IFSR
MCR p15, 0, Rd, c5, c0, 1; write IFSR
```

Register format



Register bits

Bits	Description
[31:9]	UNPREDICTABLE/SHOULD BE ZERO
[8]	Always reads as zero. Writes are ignored.
[7:4]	Specifies which of the 16 domains (D15–D0) was being accessed when a data fault occurred.
[3:0]	Type of fault generated. (See "MemoryManagement Unit (MMU)," beginning on page 105.)

Downloaded from www.Manualslib.com manuals search engine

Status and domain fields

This table shows the encodings used for the status field in the Fault Status register, and indicates whether the domain field contains valid information. See "MMU faults and CPU aborts" on page 119 for information about MMU aborts in Fault Address and Fault Status registers.

Priority	Source	Size	Status	Domain
Highest	Alignment	N/A	0b00x1	Invalid
	External abort on translation	First level	0b1100	Invalid
		Second level	0b1110	Valid
	Translation	Section page	0b0101	Invalid
			0b0111	Valid
	Domain	Section page	0b1001	Valid
			0b1011	Valid
	Permission	Section page	0b1101	Valid
			0b1111	Valid
Lowest	External abort	Section page	0b1000	Valid
			0b1010	Valid

R6: Fault Address register

Register R6 accesses the Fault Address register (FAR). The Fault Address register contains the modified virtual address of the access attempted when a data abort occurred. This register is updated only for data aborts, not for prefetch aborts; it is updated also for alignment faults and external aborts that occur while the MMU is disabled.

Writing R6 sets the Fault Address register to the value of the data written. This is useful for debugging, to restore the value of a Fault Address register to a previous state.

The CRm and opcode_2 fields SHOULD BE ZERO when reading or writing R6.

Access instructions

Use these instructions to access the Fault Address register:

```
MRC p15, 0, Rd, c6, c0, 0; read FAR
MCR p15, 0, Rd, c6, c0, 0; write FAR
```


Downloaded from www.Manualslib.com manuals search engine

R7:Cache Operations register

Register R7 controls the caches and write buffer. The function of each cache operation is selected by the opcode_2 and CRm fields in the MCR instruction that writes to CP15 R7. Writing other opcode_2 or CRm values is UNPREDICTABLE.

Reading from R7 is UNPREDICTABLE, with the exception of the two test and clean operations (see "Cache operation functions" on page 95 and "Test and clean DCache instructions" on page 96).

Write instruction

Use this instruction to write to the Cache Operations register:

```
MCR p15, opcode_1, Rd, CRn, CRm, opcode_2
```

Cache functions

This table describes the cache functions provided by register R7.

Function	Description
Invalidate cache	Invalidates all cache data, including any dirty data.
Invalidate single entry using either index or modified virtual address	Invalidates a single cache line, discarding any dirty data.
Clean single data entry using either index or modified virtual address	Writes the specified DCache line to main memory if the line is marked valid and dirty. The line is marked as not dirty, and the valid bit is unchanged.
Clean and invalidate single data entry using wither index or modified virtual address.	Writes the specified DCache line to main memory if the line is marked valid and dirty. The line is marked not valid.
Test and clean DCache	Tests a number of cache lines, and cleans one of them if any are dirty. Returns the overall dirty state of the cache in bit 30. (See "Test and clean DCache instructions" on page 96).
Test, clean, and invalidate DCache	Tests a number of cache lines, and cleans one of them if any are dirty. When the entire cache has been tested and cleaned, it is invalidated. (See "Test and clean DCache instructions" on page 96).
Prefetch ICache line	Performs an ICache lookup of the specified modified virtual address. If the cache misses and the region is cachable, a linefill is performed.

Downloaded from www.Manualslib.com manuals search engine

Function	Description
Drain write buffer	Acts as an explicit memory barrier. This instruction drains the contents of the write buffers of all memory stores occurring in program order before the instruction is completed. No instructions occurring in program order after this instruction are executed until the instruction completes. Use this instruction when timing of specific stores to the level two memory system has to be controlled (for example, when a store to an interrupt acknowledge location has to complete before interrupts are enabled).
Wait for interrupt	Drains the contents of the write buffers, puts the processor into low-power state, and stops the processor from executing further instructions until an interrupt (or debug request) occurs. When an interrupt does occur, the MCR instruction completes, and the IRQ or FIRQ handler is entered as normal. The return link in R14_irq or R14_fiq contains the address of the MCR instruction plus eight, so the typical instruction used for interrupt return (SUBS PC,R14,#4) returns to the instruction following the MCR.

Cache operation functions

This table lists the cache operation functions and associated data and instruction formats for R7.

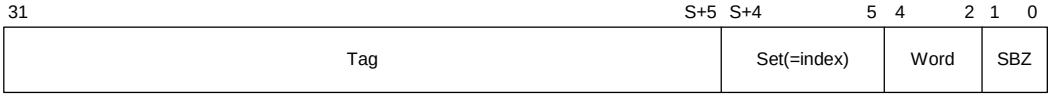
Function/operation	Data format	Instruction
Invalidate ICache and DCache	SBZ	MCR p15, 0, Rd, c7, c7, 0
Invalidate ICache	SBZ	MCR p15, 0, Rd, c7, c5, 0
Invalidate ICache single entry (MVA)	MVA	MCR p15, 0, Rd, c7, c5, 1
Invalidate ICache single entry (set/way)	Set/Way	MCR p15, 0, Rd, c7, c5, 2
Prefetch ICache line (MVA)	MVA	MCR p15, 0, Rd, c7, c13, 1
Invalidate DCache	SBZ	MCR p15, 0, Rd, c7, c6, 0
Invalidate DCache single entry (MVA)	MVA	MCR p15, 0, Rd, c7, c6, 1
Invalidate DCache single entry (set/way)	Set/Way	MCR p15, 0, Rd, c7, c6, 2
Clean DCache single entry (MVA)	MVA	MCR p15, 0, Rd, c7, c10, 1
Clean DCache single entry (set/way)	Set/Way	MCR p15, 0, Rd, c7, C10, 2
Test and clean DCache	N/A	MRC p15, 0, Rd, c7, c10, 3
Clean and invalidate DCache entry (MVA)	MVA	MCR p15, 0, Rd, c7, c14, 1
Clean and invalidate DCache entry (set/way)	Set/Way	MCR p15, 0, Rd, c7, c14, 2
Test, clean, and invalidate DCache	N/A	MRC p15, 0, Rd, c7, c14, 3

Downloaded from www.Manualslib.com manuals search engine

Function/operation	Data format	Instruction
Drain write buffer	SBZ	MCR p15, 0, Rd, c7, c10, 4
Wait for interrupt	SBZ	MCR p15, 0, Rd, c7, c0, 4

Modified virtual address format (MVA)

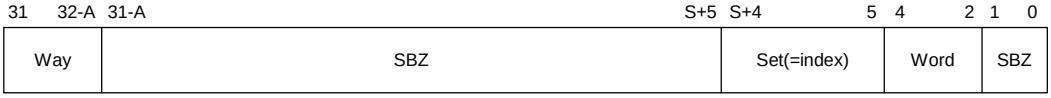
This is the modified virtual address format for Rd for the CP15 R7 MCR operations.



- The tag, set, and word fields define the MVA.
- For all cache operations, the word field SHOULD BE ZERO.

Set/Way format

This is the Set/Way format for Rd for the CP15 R7 MCR operations.



- A and S are the base-two logarithms of the associativity and the number of sets.
- The set, way, and word files define the format.
- For all of the cache operations, word SHOULD BE ZERO.

Set/Way example

For example, a 16 KB cache, 4-way set associative, 8-word line results in the following:

- A = $\log_2$ associativity = $\log_2 4 = 2$
- S = $\log_2$ NSETS where

  NSETS = cache size in bytes/associativity/line length in bytes:

  NSETS = 16384/4/32 = 128

  Result: S = $\log_2 128 = 7$

Test and clean DCache instructions

The test and clean DCache instruction provides an efficient way to clean the entire DCache, using a simple loop. The test and clean DCache instruction tests a number of lines in the DCache to determine whether any of them are dirty. If any dirty lines are found, one of those lines is cleaned. The test and clean DCache instruction also returns the status of the entire DCache in bit 30.

Downloaded from www.Manualslib.com manuals search engine

Note: The test and clean DCache instruction MRC p15, 0, r15, c7, c10, 3 is a special encoding that uses r15 as a destination operand. The PC is not changed by using this instruction, however. This MRC instruction also sets the condition code flags.

If the cache contains any dirty lines, bit 30 is set to 0. If the cache contains no dirty lines, bit 30 is set to 1. Use the following loop to clean the entire cache:

```
tc_loop:    MRC p15, 0, r15, c7, c10, 3; test and clean
            BNE tc_loop
```

Test, clean, and invalidate DCache instruction

The test, clean, and invalidate DCache instruction is the same as the test and clean DCache instruction except that when the entire cache has been cleaned, it is invalidated. Use the following loop to test, clean, and invalidate the entire DCache:

```
tci_loop:   MRC p15, 0, r15, c7, c14, 3; test clean and invalidate
            BNE tci_loop
```

R8:TLB Operations register

Register R8 is a write-only register that controls the translation lookaside buffer (TLB). There is a single TLB used to hold entries for both data and instructions. The TLB is divided into two parts:

- Set-associative
- Fully-associative

The *fully-associative* part (also referred to as the *lockdown* part of the TLB) stores entries to be locked down. Entries held in the lockdown part of the register are preserved during an invalidate-TLB operation. Entries can be removed from the lockdown TLB using an invalidate TLB single entry operation.

TLB operations

There are six TLB operations; the function to be performed is selected by the opcode_2 and CRm fields in the MCR instruction used to write register R8. Writing other opcode_2 or CRm values is UNPREDICTABLE. Reading from this register is UNPREDICTABLE.

TLB operation instructions

Use these instruction to perform TLB operations.

Operation	Data	Instruction
Invalidate set-associative TLB	SBZ	MCR p15, 0, Rd, c8, c7, 0
Invalidate single entry	SBZ	MCR p15, 0, Rd, c8, c7. 1
Invalidate set-associative TLB	SBZ	MCR p15, 0, Rd, c8, c5, 0
Invalidate single entry	MVA	MCR p15, 0, Rd, c8, c5, 1

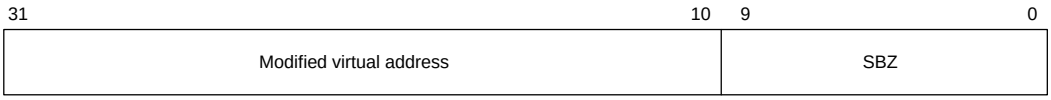
Downloaded from www.Manualslib.com manuals search engine

Operation	Data	Instruction
Invalidate set-associative TLB	SBZ	MCR p15, 0, Rd, c8, c6, 0
Invalidate single entry	MVA	MCR p15, 0, Rd, c8, c6, 1

- The *invalidate TLB operations* invalidate all the unpreserved entries in the TLB.
- The *invalidate TLB single entry operations* invalidate any TLB entry corresponding to the modified virtual address given in Rd, regardless of its preserved state. See "R10:TLB Lockdown register," beginning on page 101, for an explanation of how to preserve TLB entries.

Modified virtual address format (MVA)

This is the modified virtual address format used for invalid TLB single entry operations.



Note: If either small or large pages are used, and these pages contain subpage access permissions that are different, you must use four invalidate TLB single entry operations, with the MVA set to each subpage, to invalidate all information related to that page held in a TLB.

R9: Cache Lockdown register

Register R9 access the cache lockdown registers. Access this register using CRm = 0.

Cache ways

The Cache Lockdown register uses a cache-way-based locking scheme (format C) that allows you to control each cache way independently.

These registers allow you to control which cache-ways of the four-way cache are used for the allocation on a linefill. When the registers are defined, subsequent linefills are placed only in the specified target cache way. This gives you some control over the cache pollution cause by particular applications, and provides a traditional lockdown operation for locking critical code into the cache.

A locking bit for each cache way determines whether the normal cache allocation is allowed to access that cache way (see “Cache Lockdown register L bits” on page 99). A maximum of three cache ways of the four-way associative cache can be locked, ensuring that normal cache line replacement is performed.

Note: If no cache ways have the L bit set to 0, cache way 3 is used for all linefills.

Downloaded from www.Manualslib.com manuals search engine

Instruction or data lockdown register

The first four bits of this register determine the L bit for the associated cache way. The opcode_2 field of the MRC or MCR instruction determines whether the instruction or data lockdown register is accessed:

opcode_2=0	Selects the DCache Lockdown register, or the Unified Cache Lockdown register if a unified cache is implemented. The ARM926EJ-S processor has separate DCache and ICache.
opcode_2=1	Selects the ICache Lockdown register.

Access instructions

Use these instructions to access the CacheLockdown register.

Function	Data	Instruction
Read DCache Lockdown register	L bits	MRC p15, 0, Rd, c9, c0, 0
Write DCache Lockdown register	L bits	MCR p15, 0, Rd, c9, c0, 0
Read ICache Lockdown register	L bits	MRC p15, 0, Rd, c9, c0, 1
Write ICache Lockdown register	L bits	MCR p15, 0, Rd, c9, c0, 1

Modifying the Cache Lockdown register

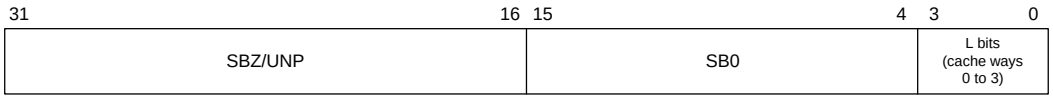
You must modify the Cache Lockdown register using a modify-read-write sequence; for example:

```
MRC p15, 0, Rn, c9, c0, 1;
ORR Rn, Rn, 0x01;
MCR p15, 0, Rn, c9, c0, 1;
```

This sequence sets the L bit to 1 for way 0 of the ICache.

Register format

This is the format for the Cache Lockdown register.



Cache Lockdown register L bits

This table shows the format of the Cache Lockdown register L bits. All cache ways are available for allocation from reset.

Bits	4-way associative	Notes
[31:16]	UNP/SBZ	Reserved
[15:4]	0xFFF	SBO

Downloaded from www.Manualslib.com manuals search engine

Bits	4-way associative	Notes
[3]	L bit for way 3	Bits [3:0] are the L bits for each cache way: 0 Allocation to the cache way is determined by the standard replacement algorithm (reset state) 1 No allocation is performed to this way
[2]	L bit for way 2	
[1]	L bit for way 1	
[0]	L bit for way 0	

Lockdown cache: Specific loading of addresses into a cache-way

Use this procedure to lockdown cache. The procedure to lock down code and data into way *i* of cache, with *N* ways, using format C, makes it impossible to allocate to any cache way other than the target cache way:

1. Ensure that no processor exceptions can occur during the execution of this procedure; for example, disable interrupts. If this is not possible, all code and data used by any exception handlers must be treated as code and data as in Steps 2 and 3.
2. If an ICache way is being locked down, be sure that all the code executed by the lockdown procedure is in an uncachable area of memory or in an already locked cache way.
3. If a DCache way is being locked down, be sure that all data used by the lockdown procedure is in an uncachable area of memory or is in an already locked cache way.
4. Ensure that the data/instructions that are to be locked down are in a cachable area of memory.
5. Be sure that the data/instructions that are to be locked down are not already in the cache. Use the Cache Operations register (R7) clean and/or invalidate functions to ensure this.
6. Write these settings to the Cache Lockdown register (R9), to enable allocation to the target cache way:

   CRm = 0

   Set L == 0 for bit i

   Set L == 1 for all other bits
7. For each of the cache lines to be locked down in cache way *i*:
   - If a DCache is being locked down, use an LDR instruction to load a word from the memory cache line to ensure that the memory cache line is loaded into the cache.
   - If an ICache is being locked down, use the Cache Operations register (R7) MCR prefetch ICache line (<CRm>==c13, <opcode2>==1) to fetch the memory cache line into the cache.

Downloaded from www.Manualslib.com manuals search engine

8 Write <CRm>==0 to Cache Lockdown register (R9), setting L==1 for bit *i* and restoring all other bits to the values they had before the lockdown routine was started.

Cache unlock procedure

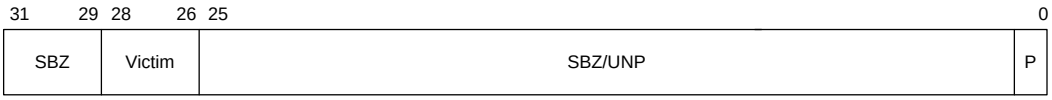
To unlock the locked down portion of the cache, write to Cache Lockdown register (R9) setting L==0 for the appropriate bit. The following sequence, for example, sets the L bit to 0 for way 0 of the ICache, unlocking way 0:

```
MRC p15, 0, Rn, c9, c0, 1;
BIC Rn, Rn, 0x01;
MCR p15, 0, Rn, c9, c0, 1;
```

R10:TLB Lockdown register

The TLB Lockdown register controls where hardware page table walks place the TLB entry — in the set associative region or the lockdown region of the TLB. If the TLB entry is put in the lockdown region, the register indicates which entry is written. The TLB lockdown region contains eight entries (see the discussion of the TLB structure in "TLB structure," beginning on page 126, for more information).

Register format



P bit

When writing the TLB Lockdown register, the value in the P bit (D0) determines in which region the TLB entry is placed:

P=0	Subsequent hardware page table walks place the TLNB entry in the set associative region of the TLB.
P=1	Subsequent hardware page table walks place the TLB entry in the lockdown region at the entry specified by the victim, in the range 0–7.

Invalidate operation

TLB entries in the lockdown region are preserved so invalidate-TLB operations only invalidate the unpreserved entries in the TLB; that is, those entries in the set-associative region. Invalidate-TLB single entry operations invalidate any TLB entry corresponding to the modified virtual address given in Rd, regardless of the entry's preserved state; that is, whether they are in lockdown or set-associative TLB regions. See "R8:TLB Operations register" on page 97 for a description of the TLB-invalidate operations.

Downloaded from www.Manualslib.com manuals search engine

Programming instructions

Use these instructions to program the TLB Lockdown register:

Function	Instruction
Read data TLB lockdown victim	MRC p15, 0, Rd, c10, c0, 0
Write data TLB lockdown victim	MCR p15, 0, Rd, c10, c0, 0

The victim automatically increments after any table walk that results in an entry being written into the lockdown part of the TLB.

Note: It is not possible for a lockdown entry to map entirely either small or large pages, unless all subpage access permissions are the same. Entries can still be written into the lockdown region, but the address range that is mapped covers only the subpage corresponding to the address that was used to perform the page table walk.

Sample code sequence

This example shows the code sequence that locks down an entry to the current victim.

```
ADR r1,LockAddr          ;   set R1 to the value of the address to be locked down
MCR p15,0,r1,c8,c7,1     ;   invalidate TLB single entry to ensure that
                         ;   LockAddr is not already in the TLB
MRC p15,0,r0,c10,c0,0    ;   read the lockdown register
ORR r0,r0,#1             ;   set the preserve bit
MCR p15,0,r0,c10,c0,0    ;   write to the lockdown register
LDR r1,[r1]              ;   TLB will miss, and entry will be loaded
MRC p15,0,r0,c10,c0,0    ;   read the lockdown register (victim will have
                         ;   incremented
BIC r0,r0,#1             ;   clear preserve bit
MCR p15,0,r0,c10,c0,0    ; write to the lockdown register
```

R11 and R12 registers

Accessing (reading or writing) these registers causes UNPREDICTABLE behavior.

R13:Process ID register

The Process ID register accesses the process identifier registers. The register accessed depends on the value on the opcode_2 field:

opcode_2=0	Selects the *Fast Context Switch Extension* (FCSE) *Process Identifier* (PID) register.

Downloaded from www.Manualslib.com manuals search engine

opcode_2=1	Selects the context ID register.

Use the Process ID register to determine the process that is currently running. The process identifier is set to 0 at reset.

FCSE PID register

Addresses issued by the ARM926EJ-S core, in the range 0 to 32 MB, are translated according to the value contained in the FCSE PID register. Address A becomes A + (FCSE PID x 32 MB); it is this modified address that the MMU and caches see. Addresses above 32 MB are not modified. The FCSE PID is a 7-bit field, which allows 128 x 32 MB processes to be mapped.

If the FCSE PID is 0, there is a flat mapping between the virtual addresses output by the ARM926EJ-S core and the modified virtual addresses used by the caches and MMU. The FCSE PID is set to 0 at system reset.

If the MMU is disabled, there is no FCSE address translation.

FCSE translation is not applied for addresses used for entry-based cache or TLB maintenance operations. For these operations, VA=MVA.

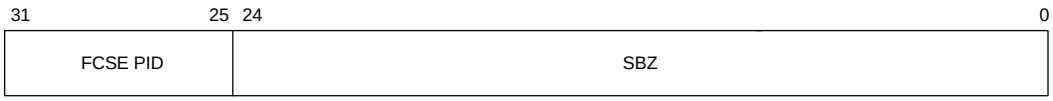
Access instructions

Use these instructions to access the FCSE PID register:

Function	Data	ARM instruction
Read FCSE PID	FCSE PID	MRC p15,0,Rd,c13,c0,0
Write FCSE PID	FCSE PID	MCR p15,0,Rd,c13,c0,0

Register format

This is the format of the FCSE PID register.



Performing a fast context switch

You can perform a fast context switch by writing to the Process ID register (R13) with opcode_2 set to 0. The contents of the caches and the TLB do not have to be flushed after a fast context switch because they still hold address tags. The two instructions after the FCSE PID has been written have been fetched with the old FCSE PID, as shown in this code example:

```
{FCSE PID = 0}
    MOV r0, #1:SHL:25          ;Fetched with FCSE PID = 0
    MCR p15,0,r0,c13,c0,0      ;Fetched with FCSE PID = 0
    A1                         ;Fetched with FCSE PID = 0
    A2                         ;Fetched with FCSE PID = 0
    A3                         ;Fetched with FCSE PID = 1
```


Downloaded from www.Manualslib.com manuals search engine

A1, A2, and A3 are the three instructions following the fast context switch.

Context ID register

The Context ID register provides a mechanism that allows real-time trace tools to identify the currently executing process in multi-tasking environments.

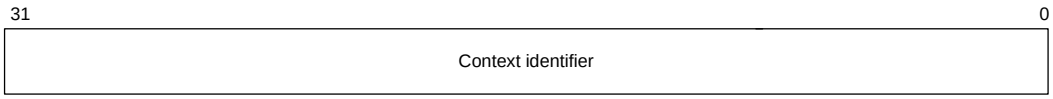
Access instructions

Use these instructions to access the Context ID register:

Function	Data	ARM instruction
Read context ID	Context ID	MRC p15,0,Rd,c13,c0,1
Write context ID	Context ID	MCR p15,0,Rd,c13,c0,1

Register format

This is the format of the Context ID register (Rd) transferred during this operation.



R14 register

Accessing (reading or writing) this register is reserved.

R15: Test and debug register

Register R15 to provides device-specific test and debug operations in ARM926EJ-S processors. Use of this register currently is reserved.

Jazelle(Java)

The ARM926EJ-S processor has ARM's embedded Jazelle Java acceleration hardware in the core. Java offers rapid application development to software engineers.

The ARM926EJ-S processor core executes an extended ARMv5TE instruction set, which includes support for Java byte code execution (ARMv5TEJ). An ARM optimized *Java Virtual Machine (JVM)* software layer has been written to work with the Jazelle hardware. The Java byte code acceleration is accomplished by the following:

- Hardware, which directly executes 80% of simple Java byte codes.

Downloaded from www.Manualslib.com manuals search engine

- Software emulation within the ARM-optimized JVM, which addresses the remaining 20% of the Java byte codes.

DSP

The ARM926EJ-S processor core provides enhanced DSP capability. Multiply instructions are processed using a single cycle 32x16 implementation. There are 32x32, 32x16, and 16x16 multiply instructions, or *Multiply Accumulate (MAC)*, and the pipeline allows one multiply to start each cycle. Saturating arithmetic improves efficiency by automatically selecting saturating behavior during execution, and is used to set limits on signal processing calculations to minimize the effect of noise or signal errors. All of these instructions are beneficial for algorithms that implement the following:

- GSM protocols
- FFT
- State space servo control

MemoryManagement Unit (MMU)

The MMU provides virtual memory features required by systems operating on platforms such as WindowsCE or Linux. A single set of two-level page tables stored in main memory control the address translation, permission checks, and memory region attributes for both data and instruction accesses. The MMU uses a single, unified *Translation Lookaside Buffer (TLB)* to cache the information held in the page tables. TLB entries can be locked down to ensure that a memory access to a given region never incurs the penalty of a page table walk.

MMU Features

- Standard ARM926EJ-S architecture MMU mapping sizes, domains, and access protection scheme.
- Mapping sizes, as follows:
  - 1 MB for sections
  - 64 KB for large pages
  - 4 KB for small pages
  - 1 KB for tiny pages
- Access permissions for large pages and small pages can be specified separately for each quarter of the page (subpage permissions).
- Hardware page table walks.

Downloaded from www.Manualslib.com manuals search engine

- Invalidate entire TLB using R8: TLB Operations register (see “R8:TLB Operations register” on page 97).
- Invalidate TLB entry selected by MVA, using R8: TLB Operations register (see “R8:TLB Operations register” on page 97).
- Lockdown of TLB entries using R10: TLB Lockdown register (see “R10:TLB Lockdown register” on page 101).

Access permissions and domains

For large and small pages, access permissions are defined for each subpage (1 KB for small pages, 16 KB for large pages). Sections and tiny pages have a single set of access permissions.

All regions of memory have an associated domain. A domain is the primary access control mechanism for a region of memory. It defines the conditions necessary for an access to proceed. The domain determines whether:

- Access permissions are used to qualify the access.
- The access is unconditionally allowed to proceed.
- The access is unconditionally aborted.

In the latter two cases, the access permission attributes are ignored.

There are 16 domains, which are configured using R3: Domain Access Control register (see “R3:Domain Access Control register” on page 91).

Translated entries

The TLB caches translated entries. During CPU memory accesses, the TLB provides the protection information to the access control logic.

When the TLB contains a translated entry for the modified virtual address (MVA), the access control logic determines whether:

- Access is permitted and an off-chip access is required — the MMU outputs the appropriate physical address corresponding to the MVA.
- Access is permitted and an off-chip access is not required — the cache services the access.
- Access is not permitted — the MMU signals the CPU core to abort.

If the TLB misses (it does not contain an entry for the MVA), the translation table walk hardware is invoked to retrieve the translation information from a translation table in physical memory. When retrieved, the translation information is written into the TLB, possible overwriting an existing value.

At reset, the MMU is turned off, no address mapping occurs, and all regions are marked as noncachable and nonbufferable.

Downloaded from www.Manualslib.com manuals search engine

MMU program accessible registers

This table shows the CP15 registers that are used in conjunction with page table descriptors stored in memory to determine MMU operation.

Register	Bits	Description
R1: Control register	M, A, S, R	Contains bits to enable the MMU (M bit), enable data address alignment checks (A bit), and to control the access protection scheme (S bit and R bit).
R2: Translation Table Base register	[31:14]	Holds the physical address of the base of the translation table maintained in main memory. This base address must be on a 16 KB boundary.
R3: Domain Access Control register	[31:0]	Comprises 16 two-bit fields. Each field defines the access control attributes for one of 16 domains (D15 to D00).
R5: Fault Status registers, IFSR and DFSR	[7:0]	Indicates the cause of a data or prefetch abort, and the domain number of the aborted access when an abort occurs. Bits [7:4] specify which of the 16 domains (D15 to D00) was being accessed when a fault occurred. Bits [3:0] indicate the type of access being attempted. The value of all other bits is UNPREDICTABLE. The encoding of these bits is shown in “Priority encoding table” on page 120).
R6: Fault Address register	[31:0]	Holds the MVA associated with the access that caused the data abort. See “Priority encoding table” on page 120 for details of the address stored for each type of fault.
R8: TLB Operations register	[31:0]	Performs TLB maintenance operations. These are either invalidating all the (unpreserved) entries in the TLB, or invalidating a specific entry.
R10: TLB Lockdown register	[28:26] and 0	Enables specific page table entries to be locked into the TLB. Locking entries in the TLB guarantees that accesses to the locked page or section can proceed without incurring the time penalty of a TLB miss. This enables the execution latency for time-critical pieces of code, such as interrupt handlers, to be minimized.

All CP15 MMU registers, except R8: TLB Operations, contain state that can be read using MRC instructions, and can be written using MCR instructions. Registers R5 (Fault Status) and R6 (Fault Address) are also written by the MMU during an abort.

Writing to R8: TLB Operations causes the MMU to perform a TLB operation, to manipulate TLB entries. This register is write-only.

Address translation

The virtual address (VA) generated by the CPU core is converted to a modified virtual address (MVA) by the FCSE (fast context switch extension) using the value held in CP15 R13: Process ID register. The MMU translates MVAs into physical addresses to access external memory, and also performs access permission checking.

Downloaded from www.Manualslib.com manuals search engine

The MMU table-walking hardware adds entries to the TLB. The translation information that comprises both the address translation data and the access permission data resides in a translation table located in physical memory. The MMU provides the logic for automatically traversing this translation table and loading entries into the TLB.

The number of stages in the hardware table walking and permission checking process is one or two. depending on whether the address is marked as a section-mapped access or a page-mapped access.

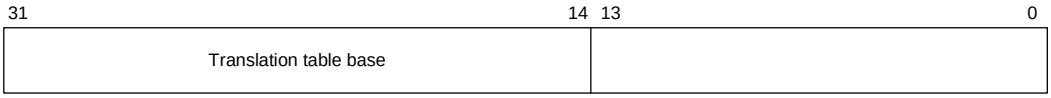
There are three sizes of page-mapped accesses and one size of section-mapped access. Page-mapped accesses are for large pages, small pages, and tiny pages.

The translation process always begins in the same way — with a level-one fetch. A section-mapped access requires only a level-one fetch, but a page-mapped access requires an additional level-two fetch.

Translation table base

The hardware translation process is initiated when the TLB does not contain a translation for the requested MVA. R2: Translation Table Base (TTB) register points to the base address of a table in physical memory that contains section or page descriptors, or both. The 14 low-order bits [13:0] of the TTB register are UNPREDICTABLE on a read, and the table must reside on a 16 KB boundary.

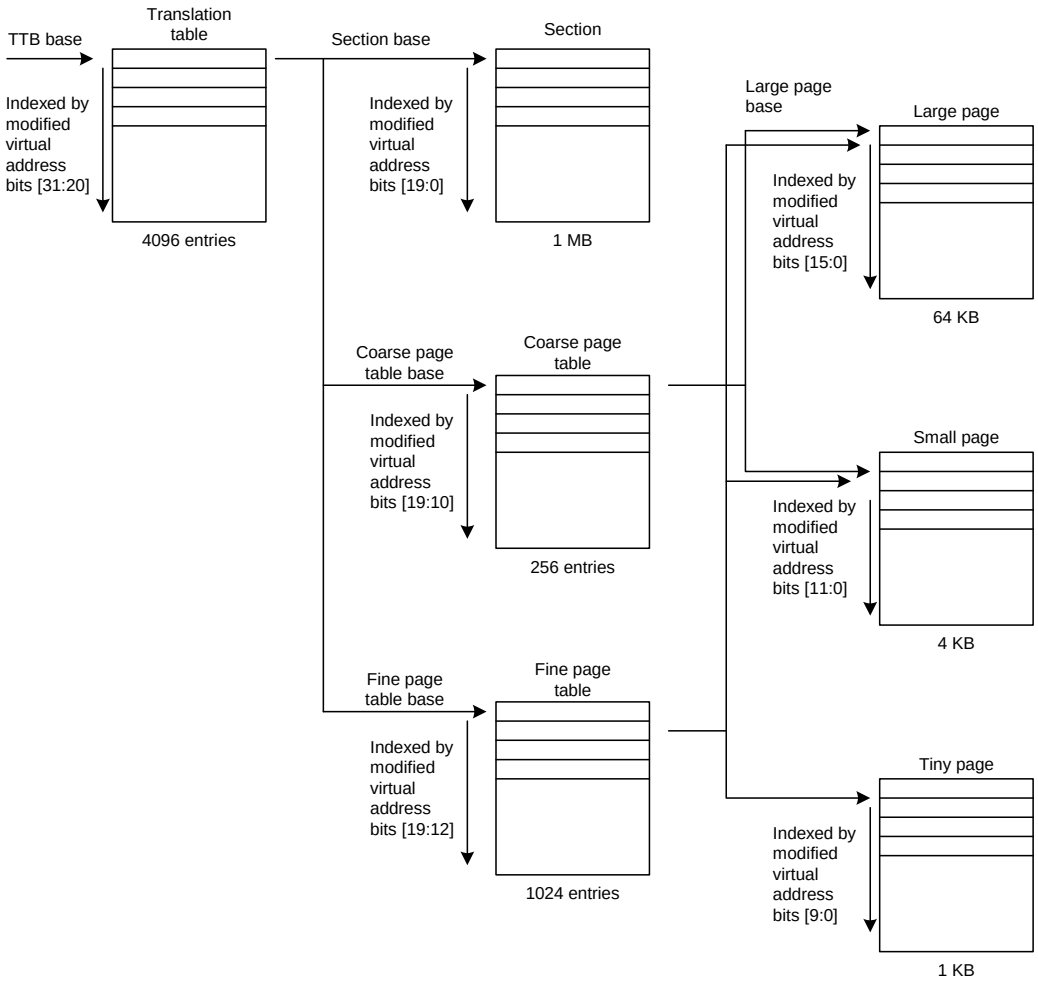
TTB register format



The translation table has up to 4096 x 32-bit entries, each describing 1 MB of virtual memory. This allows up to 4 GB of virtual memory to be addressed.

Downloaded from www.Manualslib.com manuals search engine

Table walk process

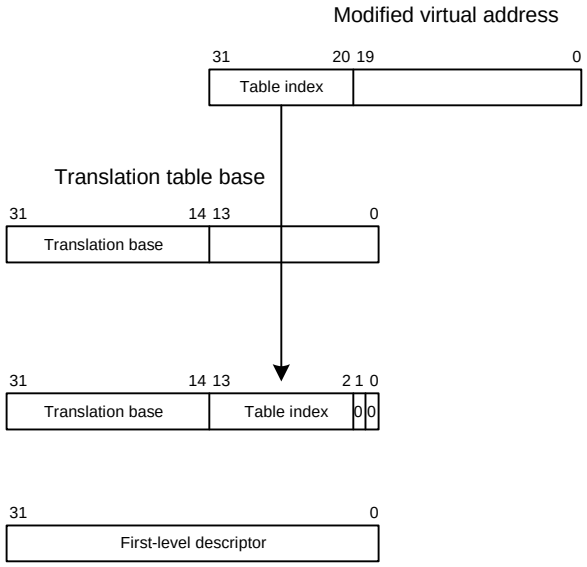


First-level fetch

Bits [31:14] of the TTB register are concatenated with bits [31:20] of the MVA to produce a 30-bit address.

Downloaded from www.Manualslib.com manuals search engine

First-level fetch concatenation and address



This address selects a 4-byte translation table entry. This is a first-level descriptor for either a section or a page.

First-level descriptor

The first-level descriptor returned is a section description, a coarse page table descriptor, a fine page table descriptor, or is invalid. This is the format of a first-level descriptor.

31 … 20	19 … 12	11	10	9	8 … 5	4	3	2	1	0	
									0	0	Fault
Coarse page table base address					Domain	1			0	1	Coarse page table
Section base address			AP		Domain	1	C	B	1	0	Section
Fine page table base address					Domain	1			1	1	Fine page table

A section descriptor provides the base address of a 1 MB block of memory.

Page table descriptors

The page table descriptors provide the base address of a page table that contains second-level descriptors. There are two page-table sizes:

- **Coarse page tables,** which have 256 entries and split the 1 MB that the table describes into 4 KB blocks.
- **Fine page tables,** which have 1024 entries and split the 1 MB that the table describes into 1 KB blocks.

Downloaded from www.Manualslib.com manuals search engine

First-level descriptor bit assignments: Priority encoding of fault status

Bits			
Section	**Coarse**	**Fine**	**Description**
[31:20]	[31:10]	[31:12]	Forms the corresponding bits of the physical address.
[19:12]	----	---	SHOULD BE ZERO
[11:10]	---	---	Access permission bits. See “Access permissions and domains” on page 106 and “Fault Address and Fault Status registers” on page 119 for information about interpreting the access permission bits.
9	9	[11:9]	SHOULD BE ZERO
[8:5]	[8:5]	[8:5]	Domain control bits
4	4	4	Must be 1.
[3:2]	---	---	Bits C and B indicate whether the area of memory mapped by this page is treated as write-back cachable, write-through cachable, noncached buffered, or noncached nonbuffered.
---	[3:2]	[3:2]	SHOULD BE ZERO
[1:0]	[1:0]	[1:0]	These bits indicate the page size and validity, and are interpreted as shown in “First-level descriptor bit assignments: Priority encoding of fault status” on page 111.

First-level descriptor bit assignments: Interpreting first level descriptor bits [1:0]

Value	Meaning	Description
0 0	Invalid	Generates a section translation fault.
0 1	Coarse page table	Indicates that this is a coarse page table descriptor.
1 0	Section	Indicates that this is a section descriptor.
1 1	Fine page table	Indicates that this is a fine page table descriptor.

Section descriptor

A section descriptor provides the base address of a 1 MB block of memory.

Section descriptor format

Downloaded from www.Manualslib.com manuals search engine

Section descriptor bit description

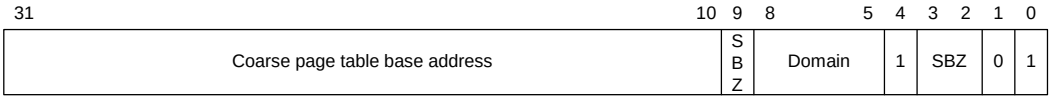
Bits	Description
[31:20]	Forms the corresponding bits of the physical address for a section.
[19:12]	Always written as 0.
[11:10]	Specify the access permissions for this section.
[09]	Always written as 0.
[8:5]	Specifies one of the 16 possible domains (held in the Domain and Access Control register) that contain the primary access controls.
4	Should be written as 1, for backwards compatibility.
[3:2]	Indicate if the area of memory mapped by this section is treated as writeback cachable, write-through cachable, noncached buffered, or noncached nonbuffered.
[1:0]	Must be *10* to indicate a section descriptor.

Coarse page table descriptor

A coarse page table descriptor provides the base address of a page table that contains second-level descriptors for either large page or small page accesses. Coarse page tables have 256 entries, splitting the 1 MB that the table describes into 4 KB blocks.

Note: If a coarse page table descriptor is returned from the first-level fetch, a second-level fetch is initiated.

Coarse page table descriptor format



Coarse page table descriptor bit description

Bits	Description
[31:10]	Forms the base for referencing the second-level descriptor (the coarse page table index for the entry derived from the MVA).
9	Always written as 0.
[8:5]	Specifies one of the 16 possible domains (held in the Domain Access Control registers) that contain the primary access controls.
4	Always written as 1.
[3:2]	Always written as 0.
[1:0]	Must be *01* to indicate a coarse page descriptor.

Fine page table descriptor

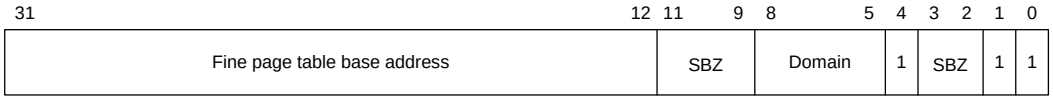
A fine page table descriptor provides the base address of a page table that contains second-level descriptors for large page, small page, or tiny page accesses. Fine

Downloaded from www.Manualslib.com manuals search engine

page tables have 1024 entries, splitting the 1 MB that the table describes into 1 KB blocks. The next two sections show the format of a fine page table descriptor and define the fine page table descriptor bit assignments.

Note: If a fine page table descriptor is returned from the first-level fetch, a second-level fetch is initiated.

Fine page table descriptor format



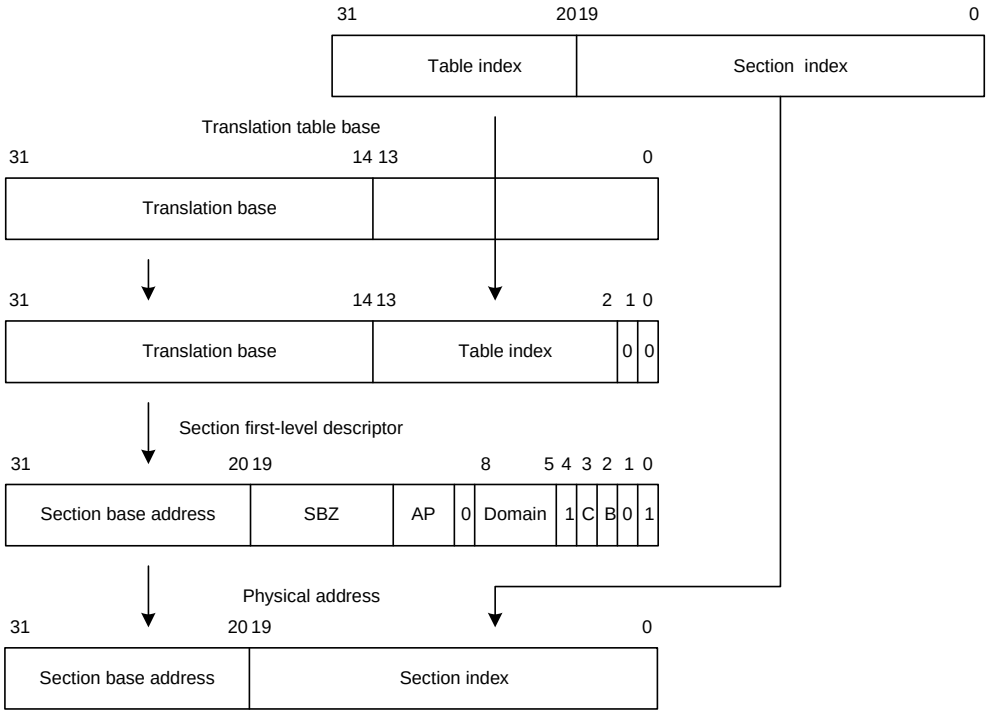
Fine page table descriptor bit description

Bits	Description
[31:12]	Forms the base for referencing the second-level descriptor (the fine page table index for the entry is derived from the MVA).
[11:9]	Always written as 0.
[8:5]	Specifies one of the 16 possible domains (held in the Domain Access Control register) that contain primary access controls.
4	Always written as 1.
[3:2}	Always written as 0.
[1:0]	Must be *11* to indicate a fine page table descriptor.

Translating section references

This figure illustrates the complete section translation sequence.

Downloaded from www.Manualslib.com manuals search engine



Second-level descriptor

The base address of the page table to be used is determined by the descriptor returned (if any) from a first-level fetch — either a coarse page table descriptor or a fine page table descriptor. The page table is then accessed and a second-level descriptor returned.

Second-level descriptor format

31 … 16	15 … 12	11	10	9	8	7	6	5	4	3	2	1	0	
												0	0	Fault
Large page base address		AP3		AP2		AP1		AP0		C	B	0	1	Large page
Small page base address		AP3		AP2		AP1		AP0		C	B	1	0	Small page
Tiny page base address								AP		C	B	1	1	Tiny page

Second-level descriptor pages

A second-level descriptor defines a tiny, small, or large page descriptor, or is invalid:

- A large page descriptor provides the base address of a 64 KB block of memory.
- A small page descriptor provides the base address of a 4 KB block of memory.

Downloaded from www.Manualslib.com manuals search engine

- A tiny page descriptor provides the base address of a 1 KB block of memory.

Coarse page tables provide base addresses for either small or large pages. Large page descriptors must be repeated in 16 consecutive entries. Small page descriptors must be repeated in each consecutive entry.

Fine page tables provide base addresses for large, small, or tiny pages. Large page descriptors must be repeated in 64 consecutive entries. Small page descriptors must be repeated in four consecutive entries. Tiny page descriptors must be repeated in each consecutive entry.

Second-level descriptor bit assignments

Bits			
Large	Small	Tiny	Description
[31:16]	[31:12]	[31:10]	Form the corresponding bits of the physical address.
[15:12]	---	[9:6]	SHOULD BE ZERO
[11:4]	[11:4]	[5:4]	Access permission bits. See “Domain access control” on page 121 and “Fault checking sequence” on page 122 for information about interpreting the access permission bits.
[3:2]	[3:2]	[3:2]	Indicate whether the area of memory mapped by this page is treated as write-back cachable, write-through cachable, noncached buffered, and noncached nonbuffered.
[1:0]	[1:0]	[1:0]	Indicate the page size and validity, and are interpreted as shown in “First-level descriptor bit assignments: Interpreting first level descriptor bits [1:0]” on page 111.

Second-level descriptor least significant bits

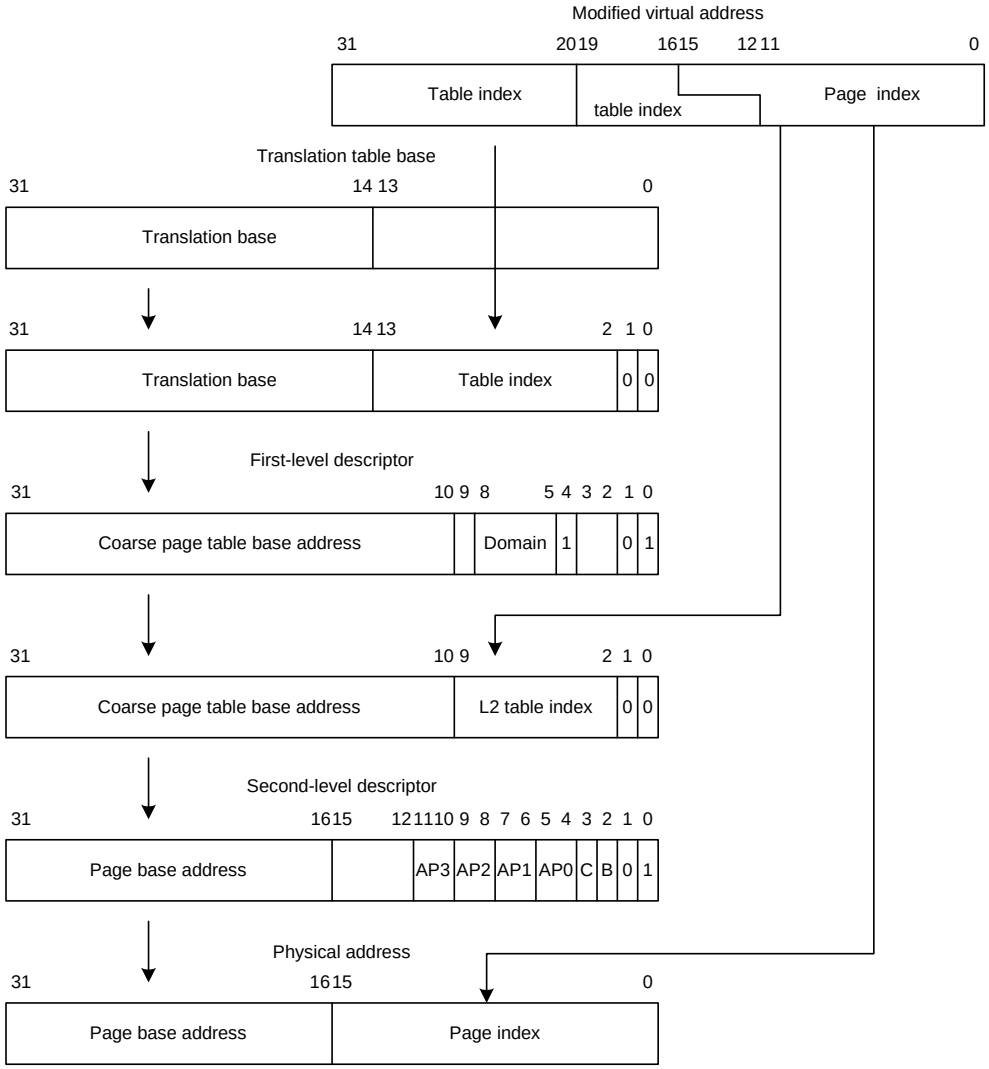
The two least significant bits of the second-level descriptor indicate the descriptor type, as shown in this table.

Value	Meaning	Description
0 0	Invalid	Generates a page translation fault.
0 1	Large page	Indicates that this is a 64 KB page.
1 0	Small page	Indicates that this is a 4 KB page.
1 1	Tiny page	Indicates that this is a 1 KB page.

Note: Tiny pages do not support subpage permissions and therefore have only one set of access permission bits.

Downloaded from www.Manualslib.com manuals search engine

Translation sequence for large page references

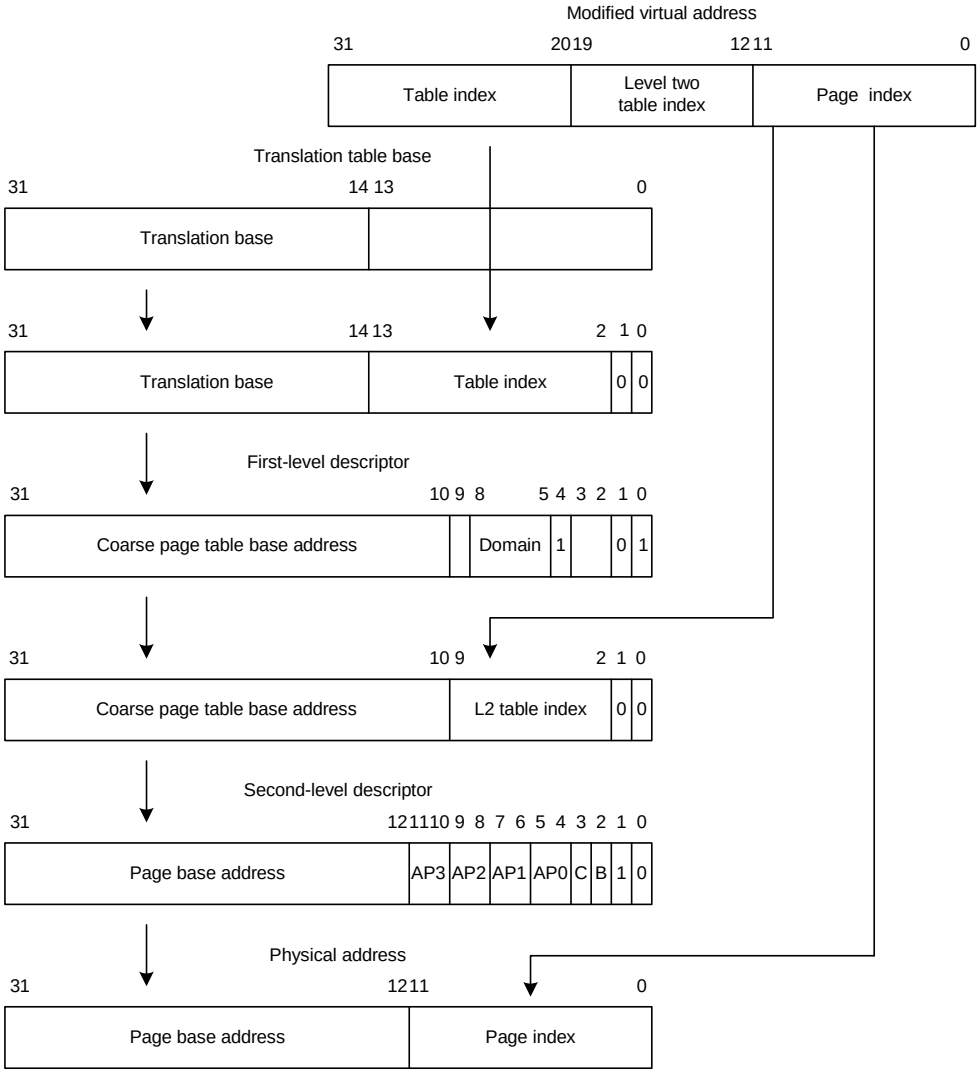


Because the upper four bits of the page index and low-order four bits of the coarse page table index overlap, each coarse page table entry for a large page must be duplicated 16 times (in consecutive memory locations) in the coarse page table.

If the large page descriptor is included in a fine page table, the high-order six bits of the page index and low-order six bits of the fine page table overlap. Each fine page table entry for a large page must be duplicated 64 times.

Downloaded from www.Manualslib.com manuals search engine

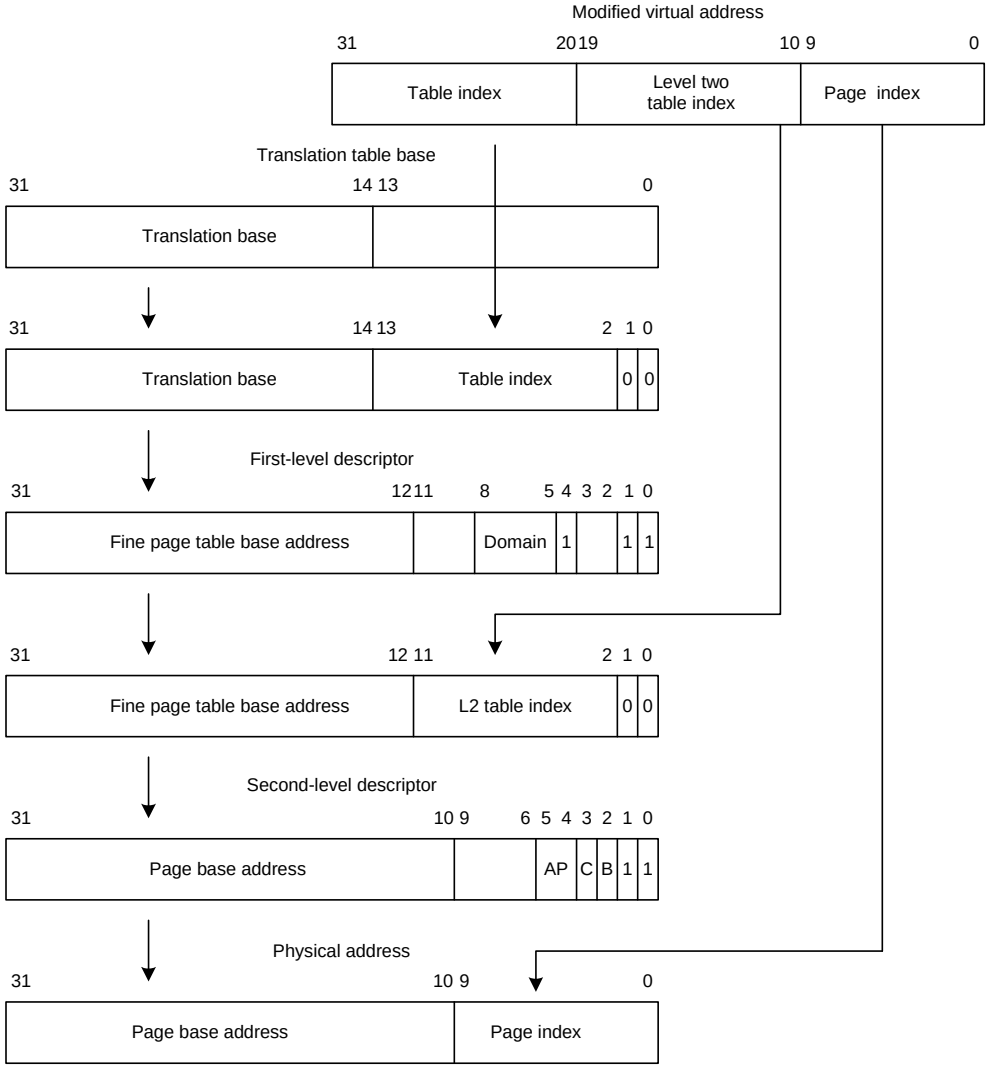
Translating sequence for small page references



If a small page descriptor is included in a fine page table, the upper two bits of the page index and low-order two bits of the fine page table index overlap. Each fine page table entry for a small page must be duplicated four times.

Downloaded from www.Manualslib.com manuals search engine

Translation sequence for tiny page references



Page translation involves one additional step beyond that of a section translation. The first-level descriptor is the fine page table descriptor; this points to the first-level descriptor.

Note: The domain specified in the first-level description and access permissions specified in the first-level description together determine whether the access has permissions to proceed. See “Domain access control” on page 121 for more information.

Subpages

You can define access permissions for subpages of small and large pages. If, during a page table walk, a small or large page has a different subpage permission, only the subpage being accessed is written into the TLB. For example, a 16 KB (large page) subpage entry is written into the TLB if the subpage permission differs, and a 64 KB entry is put in the TLB if the subpage permissions are identical.

Downloaded from www.Manualslib.com manuals search engine

When you use subpage permissions and the page entry has to be invalidated, you must invalidate all four subpages separately.

MMU faults and CPU aborts

The MMU generates an abort on these types of faults:

- Alignment faults (data accesses only)
- Translation faults
- Domain faults
- Permission faults

In addition, an external abort can be raised by the external system. This can happen only for access types that have the core synchronized to the external system:

- Page walks
- Noncached reads
- Nonbuffered writes
- Noncached read-lock-write sequence (SWP)

Alignment fault checking

Alignment fault checking is enabled by the A bit in the R1: Control register. Alignment fault checking is not affected by whether the MMU is enabled. Translation, domain, and permission faults are generated only when the MMU is enabled.

The access control mechanisms of the MMU detect the conditions that produce these faults. If a fault is detected as a result of a memory access, the MMU aborts the access and signals the fault condition to the CPU core. The MMU retains status and address information about faults generated by the data accesses in the Data Fault Status register and Fault Address register (see “Fault Address and Fault Status registers” on page 119).

The MMU also retains status about faults generated by instruction fetches in the Instruction Fault Status register.

An access violation for a given memory access inhibits any corresponding external access to the AHB interface, with an abort returned to the CPU core.

Fault Address and Fault Status registers

On a data abort, the MMU places an encoded four-bit value — the *fault status* — along with the four-bit encoded domain number in the Data Fault Status register. Similarly, on a prefetch abort, the MMU places an encoded four-bit value along with the four-bit encoded domain number in the Instruction Fault Status register. In addition, the MVA associated with the data abort is latched into the Fault Address

Downloaded from www.Manualslib.com manuals search engine

register. If an access violation simultaneously generates more than one source of abort, the aborts are encoded in the priority shown in the priority encoding table. The Fault Address register is not updated by faults caused by instruction prefetches.

Priority encoding table

Priority	Source	Size	Status	Domain
Highest	Alignment	---	0b00x1	Invalid
	External abort on transmission	First level	0b1100	Invalid
		Second level	0b1110	Valid
	Translation	Section page	0b0101	Invalid
			0b0111	Valid
	Domain	Section page	0b1001	Valid
			0b1011	Valid
	Permission	Section page	0b1101	Valid
			0b1111	Valid
Lowest	External abort	Section page	0b1000	Valid
			0b1010	Valid

Notes:

- Alignment faults can write either 0b0001 or 0b0011 into Fault Status register [3:0].
- Invalid values can occur in the status bit encoding for domain faults. This happens when the fault is raised before a valid domain field has been read from a page table description.
- Aborts masked by a higher priority abort can be regenerated by fixing the cause of the higher priority abort, and repeating the access.
- Alignment faults are not possible for instruction fetches.
- The Instruction Fault Status register can be updated for instruction prefetch operations (MCR p15,0,Rd,c7,c13,1).

Fault Address register (FAR)

For load and store instructions that can involve the transfer of more than one word (LDM/STM, STRD, and STC/LDC), the value written into the Fault Address register depends on the type of access and, for external aborts, on whether the access crosses a 1 KB boundary.

FAR values for multi-word transfers

Domain	Fault Address register
Alignment	MVA of first aborted address in transfer
External abort on translation	MVA of first aborted address in transfer
Translation	MVA of first aborted address in transfer

Downloaded from www.Manualslib.com manuals search engine

Domain	Fault Address register
Domain	MVA of first aborted address in transfer
Permission	MVA of first aborted address in transfer
External about for noncached reads, or nonbuffered writes	MVA of last address before 1KB boundary, if any word of the transfer before 1 KB boundary is externally aborted. MVA of last address in transfer if the first externally aborted word is after the 1 KB boundary.

Compatibility issues

- To enable code to be ported easily to future architectures, it is recommended that no reliance is made on external abort behavior.
- The Instruction Fault Status register is intended for debugging purposes only.

Domain access control

MMU accesses are controlled primarily through the use of domains. There are 16 domains, and each has a two-bit field to define access to it. Client users and Manager users are supported.

The domains are defined in the R3: Domain Access Control register; the register format in “R3:Domain Access Control register” on page 91 shows how the 32 bits of the register are allocated to define the 16 two-bit domains.

Specifying access permissions

This table shows how the bits within each domain are defined to specify access permissions.

Value	Meaning	Description
0 0	No access	Any access generates a domain fault.
0 1	Client	Accesses are checked against the access permission bits in the section or page descriptor.
1 0	Reserved	Reserved. Currently behaves like *no access* mode.
1 1	Manager	Accesses are not checked against the access permission bits, so a permission fault cannot be generated.

Interpreting access permission bits

This table shows how to interpret the *access permission (AP)* bits, and how the interpretation depends on the R and S bits in the R1: Control register (see "R1: Control register," beginning on page 88).

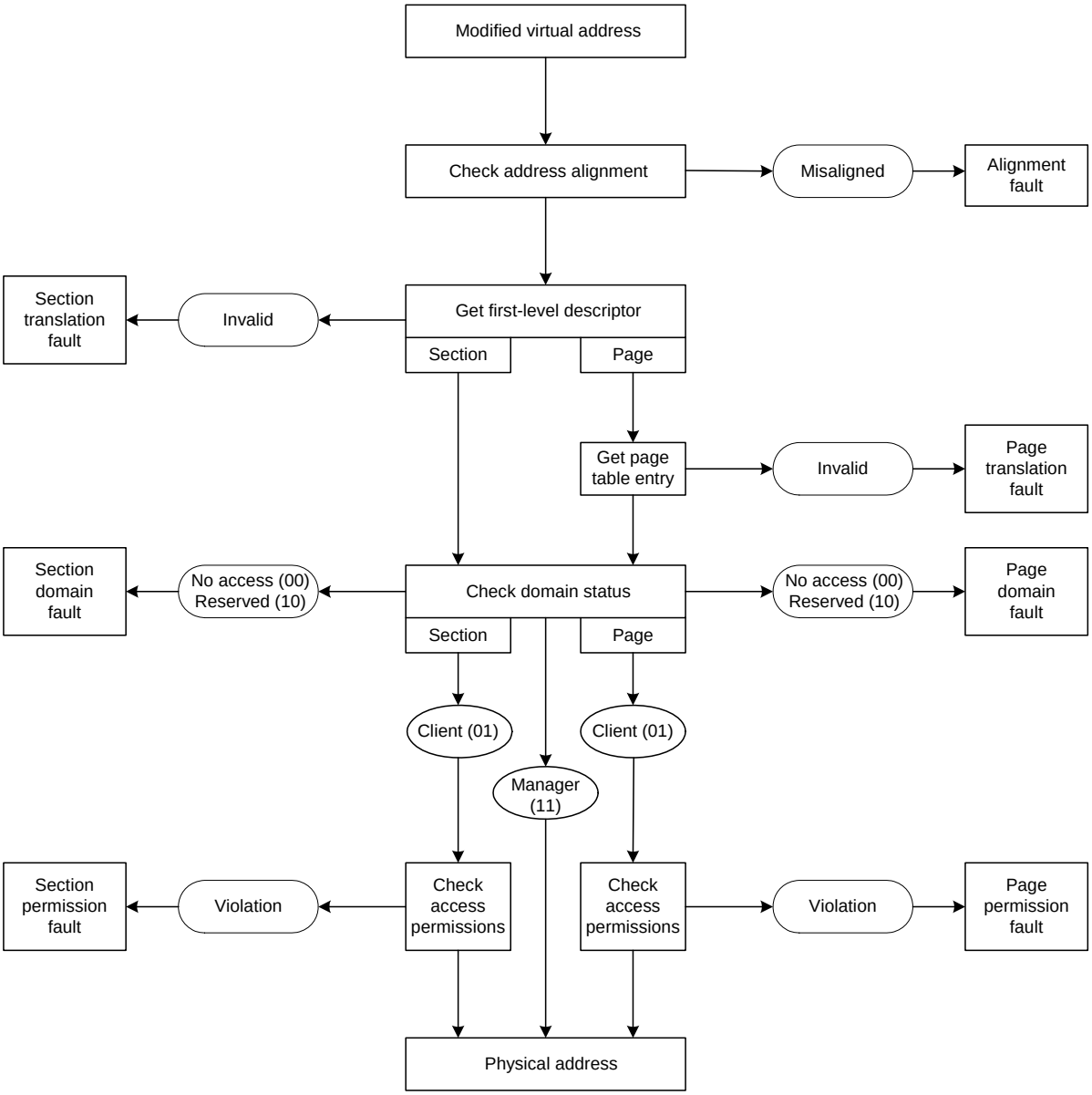
Downloaded from www.Manualslib.com manuals search engine

AP	S	R	Privileged permissions	User permissions
0 0	0	0	No access	No access
0 0	1	0	Read only	Read only
0 0	0	1	Read only	Read only
0 0	1	1	UNPREDICTABLE	UNPREDICTABLE
0 1	x	x	Read/write	No access
1 0	x	x	Read/write	Read only
1 1	x	x	Read/write	Read/write

Fault checking sequence

The sequence the MMU uses to check for access faults is different for sections and pages. The next figure shows the sequence for both types of access.

Downloaded from www.Manualslib.com manuals search engine



The conditions that generate each of the faults are discussed in the following sections.

Alignment faults

If alignment fault checking is enabled (the A bit in the R1: Control register is set; see "R1: Control register," beginning on page 88), the MMU generates an alignment fault on any data word access if the address is not word-aligned, or on any halfword access if the address is not halfword-aligned — irrespective of whether the MMU is enabled. An alignment fault is not generated on any instruction fetch or byte access.

Downloaded from www.Manualslib.com manuals search engine

Note: If an access generates an alignment fault, the access sequence aborts without reference to other permission checks.

Translation faults

There are two types of translation fault: section and page.

- A section translation fault is generated if the level one descriptor is marked as invalid. This happens if bits [1:0] of the descriptor are both 0.
- A page translation fault is generated if the level one descriptor is marked as invalid. This happens if bits [1:0] of the descriptor are both 0.

Domain faults

There are two types of domain faults: section and page.

- **Section:** The level one descriptor holds the four-bit domain field, which selects one of the 16 two-bit domains in the Domain Access Control register. The two bits of the specified domain are then checked for access permissions as described in “Interpreting access permission bits” on page 121. The domain is checked when the level one descriptor is returned.
- **Page:** The level one descriptor holds the four-bit domain field, which selects one of the 16 two-bit domains in the Domain Access Control register. The two bits of the specified domain are then checked for access permissions as described in “Interpreting access permission bits” on page 121. The domain is checked when the level one descriptor is returned.

If the specified access is either *no access* (00) or *reserved* (10), either a section domain fault or a page domain fault occurs.

Permission faults

If the two-bit domain field returns *client* (01), access permissions are checked as follows:

- **Section:** If the level one descriptor defines a section-mapped access, the AP bits of the descriptor define whether the access is allowed, per “Interpreting access permission bits” on page 121. The interpretation depends on the setting of the S and R bits (see "R1: Control register," beginning on page 88). If the access is not allowed, a section permission fault is generated.
- **Large page or small page:** If the level one descriptor defines a page-mapped access and the level two descriptor is for a large or small page, four access permission fields (AP3 to AP0) are specified, each corresponding to one quarter of the page.

  For small pages, AP3 is selected by the top 1 KB of the page and AP0 is selected by the bottom 1 KB of the page.

  For large pages, AP3 is selected by the top 16 KB of the page and AP0 is selected by the bottom 16 KB of the page. The selected AP bits are then

Downloaded from www.Manualslib.com manuals search engine

interpreted in the same way as for a section (see “Interpreting access permission bits” on page 121).

The only difference is that the fault generated is a page permission fault.

- **Tiny page:** If the level one descriptor defines a page-mapped access and the level two descriptor is for a tiny page, the AP bits of the level one descriptor define whether the access is allowed in the same way as for a section. The fault generated is a page permission fault.

External aborts

In addition to MMU-generated aborts, external aborts cam be generated for certain types of access that involve transfers over the AHB bus. These aborts can be used to flag errors on external memory accesses. Not all accesses can be aborted in this way, however.

These accesses can be aborted externally:

- Page walks
- Noncached reads
- Nonbuffered writes
- Noncached read-lock-write (SWP) sequence

For a read-lock-write (SWP) sequence, the write is always attempted if the read externally aborts.

A swap to an NCB region is forced to have precisely the same behavior as a swap to an NCNB region. This means that the write part of a swap to an NCB region can be aborted externally.

Enabling and disabling the MMU

Enabling the MMU

Before enabling the MMU using the R1: Control register, you must perform these steps:

1. Program the R2: Translation Table Base register and the R3: Domain Access Control register.
2. Program first-level and second-level page tables as required, ensuring that a valid translation table is placed in memory at the location specified by the Translation Table Base register.

When these steps have been performed, you can enable the MMU by setting R1: Control register bit 0 (the M bit) to high.

Downloaded from www.Manualslib.com manuals search engine

Care must be taken if the translated address differs from the untranslated address, because several instructions following the enabling of the MMU might have been prefetched with MMU off (VA=MVA=PA). If this happens, enabling the MMU can be considered as a branch with delayed execution. A similar situation occurs when the MMU is disabled. Consider this code sequence:

```
MRC p15, 0, R1, c1, C0, 0        ; Read control register
ORR R1, #0x1                     ; Set M bit
MCR p15, 0,R1,C1, C0,0           ; Write control register and enable MMU
Fetch Flat
Fetch Flat
Fetch Translated
```

Note: Because the same register (R1: Control register) controls the enabling of ICache, DCache, and the MMU, all three can be enabled using a single MCR instruction.

Disabling the MMU

Clear bit 0 (the M bit) in the R1: Control register to disable the MMU.

Note: If the MMU is enabled, then disabled, then subsequently re-enabled, the contents of the TLB are preserved. If these are now invalid, the TLB must be invalidated before re-enabling the MMU (see “R8:TLB Operations register” on page 97).

TLB structure

The MMU runs a single unified TLB used for both data accesses and instruction fetches. The TLB is divided into two parts:

- An eight-entry fully-associative part used exclusively for holding locked down TLB entries.
- A set-associative part for all other entries.

Whether an entry is placed in the set-associative part or lockdown part of the TLB depends on the state of the TLB Lockdown register when the entry is written into the TLB (see “R10:TLB Lockdown register” on page 101).

When an entry has been written into the lockdown part of the TLB, it can be removed only by being overwritten explicitly or, when the MVA matches the locked down entry, by an MVA-based TLB invalidate operation.

The structure of the set-associative part of the TLB does not form part of the programmer’s model for the ARM926EJ-S processor. No assumptions must be made

Downloaded from www.Manualslib.com manuals search engine

about the structure, replacement algorithm, or persistence of entries in the set-associative part — specifically:

- Any entry written into the set-associative part of the TLB can be removed at any time. The set-associative part of the TLB must be considered as a temporary cache of translation/page table information. No reliance must be placed on an entry residing or not residing in the set-associative TLB unless that entry already exists in the lockdown TLB. The set-associative part of the TLB can contain entries that are defined in the page tables but do not correspond to address values that have been accessed since the TLB was invalidated.
- The set-associative part of the TLB must be considered as a cache of the underlying page table, where memory coherency must be maintained at all times. To guarantee coherency if a level one descriptor is modified in main memory, either an invalidate-TLB or Invalidate-TLB-by-entry operation must be used to remove any cached copies of the level one descriptor. This is required regardless of the type of level one descriptor (section, level two page reference, or fault).
- If any of the subpage permissions for a given page are different, each of the subpages are treated separately. To invalidate all entries associated with a page with subpage permissions, four MVA-based invalidate operations are required — one for each subpage.

Caches and write buffer

The ARM926EJ-S processor includes an instruction cache (ICache), data cache (DCache), and write buffer. The instruction cache is 8 KB in length, and the data cache is 4 KB in length.

Cache features

- The caches are virtual index, virtual tag, addressed using the modified virtual address (MVA). This avoids cache cleaning and/or invalidating on context switch.
- The caches are four-way set associative, with a cache line length of eight words per line (32 bytes per line), and with two dirty bits in the DCache.
- The DCache supports write-through and write-back (copyback) cache operations, selected by memory region using the C and B bits in the MMU translation tables.
- The caches support *allocate on read-miss*. The caches perform critical-word first cache refilling.

Downloaded from www.Manualslib.com manuals search engine

- The caches use pseudo-random or round-robin replacement, selected by the RR bit in R1: Control register.
- Cache lockdown registers enable control over which cache ways are used for allocation on a linefill, providing a mechanism for both lockdown and controlling cache pollution.
- The DCache stores the *Physical Address Tag* (PA tag) corresponding to each DCache entry in the tag RAM for use during cache line write-backs, in addition to the virtual address tag stored in the tag RAM. This means that the MMU is not involved in DCache write-back operations, which removes the possibility of TLB misses to the write-back address.
- Cache maintenance operations provide efficient invalidation of:
  - The entire DCache or ICache
  - Regions of the DCache or ICache
  - Regions of virtual memory

  Cache maintenance operations also provide for efficient cleaning and invalidation of:
  - The entire DCache
  - Regions of the DCache
  - Regions of virtual memory

  The latter allows DCache coherency to be efficiently maintained when small code changes occur; for example, for self-modifying code and changes to exception vectors.

Write buffer

The write buffer is used for all writes to a noncachable bufferable region, write-through region, and write misses to a write-back region. A separate buffer is incorporated in the DCache for holding write-back data for cache line evictions or cleaning of dirty cache lines.

- The main write buffer has a 16-word data buffer and a four-address buffer.
- The DCache write-back buffer has eight data word entries and a single address entry.

The MCR drain write buffer instruction enables both write buffers to be drained under software control.

The MCR wait -for-interrupt causes both write buffers to be drained, and the ARM926EJ-S processor to be put into low-power state until an interrupt occurs.

Enabling the caches

On reset, the ICache and DCache entries all are invalidated and the caches disabled. The caches are not accessed for reads or writes. The caches are enabled using the I, C, and M bits from the R1: Control register, and can be enabled independently of one another.

Downloaded from www.Manualslib.com manuals search engine

ICache I and M bit settings

This table gives the I and M bit settings for the ICache, and the associated behavior.

R1 I bit	R1 M bit	ARM926EJ-S behavior
0	-----	ICache disabled. All instruction fetches are fetched from external memory (AHB).
1	0	ICache enabled, MMU disabled. All instruction fetches are cachable, with no protection checks. All addresses are flat-mapped; that is, VA=MVA=PA.
1	1	ICache enabled, MMU enabled. Instruction fetches are cachable or noncachable, depending on the page descriptor C bit (see "ICache page table C bit settings" on page 129), and protection checks are performed. All addresses are remapped from VA to PA, depending on the page entry; that is, the VA is translated to MVA and the MVA is remapped to a PA.

ICache page table C bit settings

This table shows the page table C bit settings for the ICache (R1 I bit = M bit = 1).

Page table C bit	Description	ARM926EJ-S behavior	
0	Noncachable	ICache disabled. All instruction fetches are fetched from external memory.	
1	Cachable	**Cache hit**	Read from the ICache.
		Cache miss	Linefill from external memory.

R1 register C and M bits for DCache

This table gives the R1: Control register C and M bit settings for DCache, and the associated behavior.

R1 C bit	R1 M bit	ARM926EJ-S behavior
0	0	DCache disabled. All data accesses are to the external memory.
1	0	DCache enabled, MMU disabled. All data accesses are noncachable, nonbufferable, with no protection checks. All addresses are flat-mapped; that is, VA=MVA=PA.
1	1	DCache enabled, MMU enabled. All data accesses are cachable or noncachable, depending on the page descriptor C bit and B bit (see "DCache page table C and B settings" on page 129), and protection checks are performed. All addresses are remapped from VA to PA, depending on the MMU page table entry; that is, the VA is translated to an MVA and the MVA is remapped to a PA.

DCache page table C and B settings

This table gives the page table C and B bit settings for the DCache (R1: Control register C bit = M bit = 1), and the associated behavior.

Downloaded from www.Manualslib.com manuals search engine

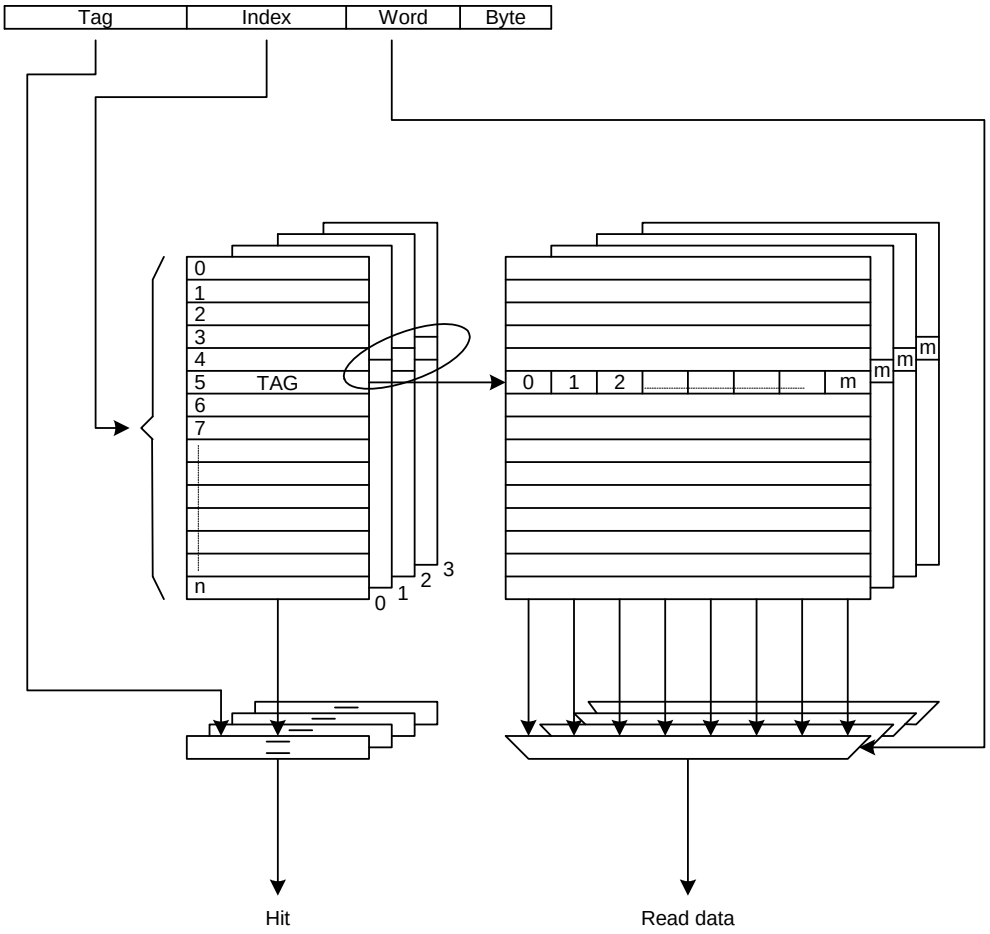
Page table C bit	Page table B bit	Description	ARM926EJ-S behavior
0	0	Noncachable, nonbufferable	DCache disabled. Read from external memory. Write as a nonbuffered store(s) to external memory. DCache is not updated.
0	1	Noncachable, bufferable	DCache disabled. Read from external memory. Write as a buffered store(s) to external memory. DCache is not updated.
1	0	Write-through	DCache enabled: **Read hit** Read from DCache. **Read miss** Linefill. **Write hit** Write to the DCache, and buffered store to external memory. **Write miss** Buffered store to external memory.
1	1	Write-back	DCache enabled: **Read hit** Read from DCache. **Read miss** Linefill. **Write hit** Write to the DCache only. **Write miss** Buffered store to external memory.

Cache MVA and Set/Way formats

This section shows how the MVA and set/way formats of ARM926EJ-S caches map to a generic virtually indexed, virtually addressed cache, shown next. The next figure shows a generic, virtually indexed, virtually addressed cache.

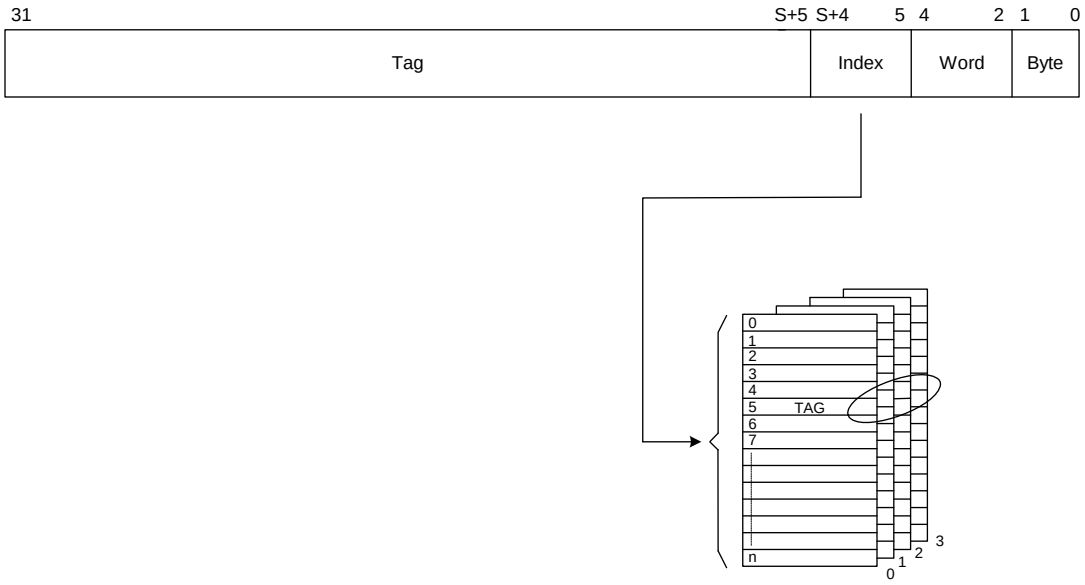
Downloaded from www.Manualslib.com manuals search engine

Generic, virtually indexed, virtually addressed cache



Downloaded from www.Manualslib.com manuals search engine

ARM926EJ-S cache format



ARM926EJ-S cache associativity

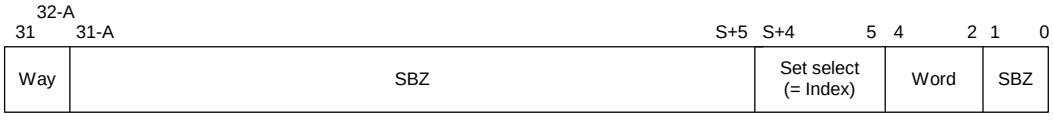
The following points apply to the ARM926EJ-S cache associativity:

- The group of tags of the same index defines a *set*.
- The number of tags in a set is the *associativity*.
- The ARM926EJ-S caches are 4-way associative.
- The range of tags addressed by the index defines a *way*.
- The number of tags is a way is the number of sets, *NSETS*.

This table shows values of S and NSETS for an ARM926EJ-S cache.

ARM926EJ-S	S	NSETS
4 KB	5	32
8 KB	6	64
16 KB	7	128
32 KB	8	256
64 KB	9	512
128 KB	10	1024

Set/way/word format for ARM926EJ-S caches



Downloaded from www.Manualslib.com manuals search engine

In this figure:

$A = \log_2$ associativity

For example, with a 4-way cache A = 2:

$S = \log_2$ NSETS

Noncachable instruction fetches

The ARM926EJ-S processor performs speculative noncachable instruction fetches to increase performance. Speculative instruction fetching is enabled at reset.

Note: It is recommended that you use ICache rather than noncachable code, when possible. Noncachable code previously has been used for operating system boot loaders and for preventing cache pollution. ICache, however, can be enabled without the MMU being enabled, and cache pollution can be controlled using the cache lockdown register.

Self-modifying code

A four-word buffer holds speculatively fetched instructions. Only sequential instructions are fetched speculatively; if the ARM926EJ-S issues a nonsequential instruction fetch, the contents of the buffer are discarded (flushed). In situations on which the contents of the prefetch buffer might become invalid during a sequence of sequential instruction fetches by the processor core (for example, turning the MMU on or off, or turning on the ICache), the prefetch buffer also is flushed. This avoids the necessity of performing an explicit *Instruction Memory Barrier* (IMB) operation, except when self-modifying code is used. Because the prefetch buffer is flushed when the ARM926EJ-S core issues a nonsequential instruction fetch, a branch instruction (or equivalent) can be used to implement the required IMB behavior, as shown in this code sequence:

```
LDMIA     R0,{R1-R5}          ; load code sequence into R1-R5
ADR       R0,self_mod_code
STMIA     R0,{R1-R5}          ; store code sequence (nonbuffered region)
B         self_mod_code       ; branch to modified code
self_mod_code:
```

This IMB application applies only to the ARM926EJ-S processor running code from a noncachable region of memory. If code is run from a cachable region of memory, or a different device is used, a different IMB implementation is required. IMBs are discussed in "Instruction Memory Barrier," beginning on page 134.

Downloaded from www.Manualslib.com manuals search engine

AHB behavior

If instruction prefetching is disabled, all instruction fetches appear on the AHB interface as single, nonsequential fetches.

If prefetching is enabled, instruction fetches appear either as bursts of four instructions or as single, nonsequential fetches. No speculative instruction fetching is done across a 1 KB boundary.

All instruction fetches, including those made in Thumb state, are word transfers (32 bits). In Thumb state, a single-word instruction fetch reads two Thumb instructions and a four-word burst reads eight instructions.

Instruction Memory Barrier

Whenever code is treated as data — for example, self-modifying code or loading code into memory — a sequence of instructions called an *instruction memory barrier (IMB)* operation must be used to ensure consistency between the data and instruction streams processed by the ARM926EJ-S processor.

Usually the instruction and data streams are considered to be completely independent by the ARM926EJ-S processor memory system, and any changes in the data side are not automatically reflected in the instruction side. For example, if code is modified in main memory, ICache may contain stale entries. To remove these stale entries, part of all of the ICache must be invalidated.

IMB operation

Use this procedure to ensure consistency between data and instruction sides:

1. **Clean the DCache.** If the cache contains cache lines corresponding to write-back regions of memory, it might contain dirty entries. These entries must be cleaned to make external memory consistent with the DCache. If only a small part of the cache has to be cleaned, it can be done by using a sequence of clean DCache single entry instructions. If the entire cache has to be cleaned, you can use the test and clean operation (see "R7:Cache Operations register," beginning on page 94).
2. **Drain the write buffer.** Executing a drain write buffer causes the ARM926EJ-S core to wait until outstanding buffered writes have completed on the AHB interface. This includes writes that occur as a result of data being written back to main memory because of clean operations, and data for store instructions.
3. **Synchronize data and instruction streams in level two AHB systems.** The level two AHB subsystem might require synchronization between data and instruction sides. It is possible for the data and instruction AHB masters to be attached to different AHB subsystems. Even if both masters are present on the same bus, some form of separate ICache might exist for performance reasons; this must be invalidated to ensure consistency.

   The process of synchronizing instructions and data in level two memory must be invoked using some form of fully blocking operation, to ensure that the end of the operation can be determined using software. It is

Downloaded from www.Manualslib.com manuals search engine

recommended that either a nonbuffered store (STR) or a noncached load (LDR) be used to trigger external synchronization.

4 **Invalidate the cache.** The ICache must be invalidated to remove any stale copies of instructions that are no longer valid. If the ICache is not being used, or the modified regions are not in cachable areas of memory, this step might not be required.

5 **Flush the prefetch buffer.** To ensure consistency, the prefetch buffer should be flushed before self-modifying code is executed (see "Self-modifying code" on page 133).

Sample IMB sequences

These sequences correspond to steps 1–4 in "IMB operation."

```
clean loop
        MRC p15, 0, r15, c7, c10, 3     ; clean entire dcache using test and clean
        BNE clean_loop

        MRC p15, 0, r0, c7, c10, 4      ; drain write buffer
        STR rx,[ry]                         ; nonbuffered store to signal L2 world to
                                            ; synchronize
        MCR p15, 0, r0, c7, c5, 0       ; invalidate icache
```

This next sequence illustrates an IMB sequence used after modifying a single instruction (for example, setting a software breakpoint), with no external synchronization required:

```
STR rx,[ry]                     ; store that modifies instruction at address ry
MCR p15, 0, ry, c7, c10, 1      ; clean dcache single entry (MVA)
MCR p15, 0, r0, c7, c10, 4      ; drain write buffer
MCR p15, 0, ry, c7, c5, 1       ; invalidate icache single entry (MVA)
```


Downloaded from www.Manualslib.com manuals search engine

Downloaded from www.Manualslib.com manuals search engine

System Control Module

CHAPTER 4

The System Control Module configures and oversees system operations for the processor, and defines both the AMBA High-speed Bus (AHB) arbiter system and system memory address space.

Features

The System Control Module uses the following to configure and maintain system operations:

- AHB arbiter system
- System-level address decoding
- 11 programmable timers
  - Watchdog timer
  - 10 general purpose timers/counters
- Interrupt controller
- Multiple configuration and status registers
- System Sleep/Wake-up processor

Bus interconnection

The AMBA AHB bus protocol uses a central multiplexor interconnection scheme. All bus masters generate the address and control signals that indicate the transfer that the bus masters want to perform. The arbiter determines which master has its address and control signals routed to all slaves. A central decoder is required to control the read data and response multiplexor, which selects the appropriate signals from the slave that is involved in the transfer.

Downloaded from www.Manualslib.com manuals search engine

System bus arbiter

The bus arbitration mechanism ensures that only one bus master has access to the system bus at any time. If you are using a system in which bus bandwidth allocation is critical, you must be sure that your worst-case bus bandwidth allocation goals can be met. See “Arbiter configuration example” on page 140 for information about configuring the AHB arbiter.

High speed bus system

The high-speed bus system is split into two subsystems:

- **High-speed peripheral subsystem:** Connects all high-speed peripheral devices to a port on the external memory controller.
- **CPU subsystem:** Connects the CPU directly to a second port on the external memory controller.

High-speed bus arbiters

The high-speed bus contains two arbiters: one for the ARM926 (CPU) and one for the main bus.

- **CPU arbiter.** Splits the bandwidth 50–50 between the data and instruction interfaces. If the CPU access is to external memory, no further arbitration is necessary; the CPU has immediate access to external memory through slave port 0 on the memory controller. If CPU access is to one of the peripherals on the main bus, however, the main arbiter will arbitrate the access.
- **Main arbiter.** Contains a 16-entry Bus Request Configuration (BRC) register. Each BRC entry represents a bus request and grant channel. Each request/grant channel can be assigned to only one bus master at a time. Each bus master can be connected to multiple request/grant channels simultaneously, however, depending on the bus bandwidth requirement of that master.

  Each request/grant channel has a two-bit Bandwidth Reduction Field (BRF) to determine how often each channel can arbitrate for the system bus — 100%, 75%, 50%, or 25%. A BRF value of 25%, for example, causes a channel to be skipped every 3 or 4 cycles. The BRC gates the bus requesting signals going into a 16-entry Bus Request register (BRR). As a default, unassigned channels in the BRC block the corresponding BRR entries from being set by any bus request signals. On powerup, only the CPU is assigned to one of the channels with 100% bandwidth strength as the default setting.

How the bus arbiter works

1. The arbiter evaluates the BRR at every bus clock until one or more bus requests are registered.

Downloaded from www.Manualslib.com manuals search engine

2 The arbiter stops evaluating the BRR until a bus grant is issued for the previous evaluation cycle.

3 The arbiter grants the bus to requesting channels, in a round-robin manner, at the rising clock edge of the last address issued for the current transaction (note that each transaction may have multiple transfers), when a SPLIT response is sampled by the arbiter, or when the bus is idling.

4 Each master samples the bus grant signal (hgrant_x) at the end of the current transfer, as indicated by the hready signal. The bus master takes ownership of the bus at this time.

5 The arbiter updates the hmaster [3:0] signals at the same time to indicate the current bus master and to enable the new master's address and control signals to the system bus.

See your AMBA standards documentation for detailed information and illustrations of AMBA AHB transactions.

Ownership

Ownership of the data bus is delayed from ownership of the address/control bus. When hready indicates that a transfer is complete, the master that owns the address/control bus can use the data bus — and continues to own that data bus — until the transaction completes.

Note: If a master is assigned more than one request/grant channel, these channels need to be set and reset simultaneously to guarantee that a non-requesting master will not occupy the system bus.

Locked bus sequence

The arbiter observes the hlock_x signal from each master to allow guaranteed back-to-back cycles, such as read-modified-write cycles. The arbiter ensures that no other bus masters are granted the bus until the locked sequence has completed. To support SPLIT or RETRY transfers in a locked sequence, the arbiter retains the bus master as granted for an additional transfer to ensure that the last transfer in the locked sequence completed successfully.

If the master is performing a locked transfer and the slave issues a split response, the master continues to be granted the bus until the slave finishes the SPLIT response. (This situation degrades AHB performance.)

Relinquishing the bus

When the current bus master relinquishes the bus, ownership is granted to the next requester.

- If there are no new requesters, ownership is granted to the last master.
- Bus parking must be maintained if other masters are waiting for SPLIT transfers to complete.

Downloaded from www.Manualslib.com manuals search engine

- If the bus is granted to a default master and continues to be in the IDLE state longer than a specified period of time, an AHB bus arbiter timeout is generated. An AHB bus arbiter timeout can be configured to interrupt the CPU or to reset the chip.

SPLIT transfers

A SPLIT transfer occurs when a slave is not ready to perform the transfer. The slave splits, or masks, its master, taking away the master's bus ownership and allowing other masters to perform transactions until the slave has the appropriate resources to perform its master's transaction.

The bus arbiter supports SPLIT transfers. When a SPLIT response is issued by a slave, the current master is masked for further bus requesting until a corresponding hsplit_x[15:0] signal is issued by the slave indicating that the slave is ready to complete the transfer. The arbiter uses the hsplit_x[15:0] signals to unmask the corresponding master, and treats the master as the highest-priority requester for the immediate next round of arbitration. The master eventually is granted access to the bus to try the transfer again.

Note: The arbiter automatically blocks bus requests with addresses directed at a "SPLITting" slave until that SPLIT transaction is completed.

Arbiter configuration example

This example shows how to configure the AHB arbiter to guarantee bandwidth to a given master. These are the conditions in this example:

- 5 AHB masters — CPU, Ethernet Rx, Ethernet Tx, IO hub, and external DMA
- AHB clock frequency — 75 MHz
- Average access time per 16-byte memory access — 4 clock cycles
- The ARM926EJ-S is guaranteed one-half the total memory bandwidth

In this example, the bandwidth for each master can be calculated using this formula:

Bandwidth per master:

= [(75MHz/2) / (4 clock cycles per access x 5 masters)] x 16 bytes

= 60MB/master

Note: The worst case scenario is that there are 90 Mbps total to be split by all 5 masters.

if this meets the requirements of all the masters, the AHB arbiter is programmed like this:

BRC0[31:24]	= 8'b1_0_00_0000	channel enabled, 100%, ARM7EJ-S
BRC0[23:16]	= 8'b1_0_00_0001	channel enabled, 100%, Ethernet Rx
BRC0[15:8]	= 8'b1_0_00_0000	channel enabled, 100%, Ethernet TX
BRC0[7:0]	= 8'b1_0_00_0101	channel enabled, 100%, IO hub
BRC1[31:24]	= 8'b1_0_00_0011	channel enabled, 100%, Ext DMA

Downloaded from www.Manualslib.com manuals search engine

BRC1[23:16]	= 8'b1_0_00_0000	channel disabled
BRC1[15:8]	= 8'b1_0_00_0000	channel disabled
BRC1[7:0]	= 8'b1_0_00_0000	channel disabled
BRC2[31:24]	= 8'b0_0_00_0000	channel disabled
BRC2[23:16]	= 8'b0_0_00_0000	channel disabled
BRC2[15:8]	= 8'b0_0_00_0000	channel disabled
BRC2[7:0]	= 8'b0_0_00_0000	channel disabled
BRC3[31:24]	= 8'b0_0_00_0000	channel disabled
BRC3[23:16]	= 8'b0_0_00_0000	channel disabled
BRC3[15:8]	= 8'b0_0_00_0000	channel disabled
BRC[7:0]	= 8'b0_0_00_0000	channel disabled

Address decoding

A central address decoder provides a select signal — hsel_x — for each slave on the bus.

This table shows how the system memory address is set up to allow access to the internal and external resources on the system bus. Note that the external memory chip select ranges can be reset after powerup. The table shows the default powerup values; you can change the ranges by writing to the BASE and MASK registers (see "System Memory Chip Select 0 Dynamic Memory Base and Mask registers" on page 190 through "System Memory Chip Select 3 Dynamic Memory Base and Mask registers" on page 193 for more information).

Address range	Size	System functions
0x0000 0000 – 0x0FFF FFFF	256 MB	System memory chip select 0 Dynamic memory (default)
0x1000 0000 – 0x1FFF FFFF	256 MB	System memory chip select 1 Dynamic memory (default)
0x2000 0000 – 0x2FFF FFFF	256 MB	System memory chip select 2 Dynamic memory (default)
0x3000 0000 – 0x3FFF FFFF	256 MB	System memory chip select 3 Dynamic memory (default)
0x4000 0000 – 0x4FFF FFFF	256 MB	System memory chip select 0 Static memory (default)
0x5000 0000 – 0x5FFF FFFF	256 MB	System memory chip select 1 Static memory (default)
0x6000 0000 – 0x6FFF FFFF	256 MB	System memory chip select 2 Static memory (default)

Downloaded from www.Manualslib.com manuals search engine

Address range	Size	System functions
0x7000 0000 – 0x7FFF FFFF	256 MB	System memory chip select 3 Static memory (default)
0x8000 0000 – 0x8FFF FFFF	256 MB	Reserved
0x9000 0000 – 0x9FFF FFFF	256 MB	IO hub
0xA000 0000 – 0xA05F FFFF	6 MB	Reserved
0xA060 0000 – 0xA06F FFFF	1 MB	Ethernet Communication Module
0xA070 0000 – 0xA07F FFFF	1 MB	Memory controller
0xA080 0000 – 0xA08F FFFF	1 MB	External DMA module
0xA090 0000 – 0xA09F FFFF	1 MB	System Control Module
0xA0A0 0000 – 0xFFFF FFFF	1526MB	Reserved

This table shows the hmaster[3:0] assignments for the processor.

Master Name	hmaster[3:0] assignment
ARM926 data	0000
Ethernet Rx	0001
Ethernet Tx	0010
IO hub	0100
ARM926 instruction	0101

Programmable timers

The processor provides 11 programmable timers:

- Software watchdog timer
- 10 general purpose timers

Software watchdog timer

The software watchdog timer, set to specific time intervals, handles gross system misbehaviors. The watchdog timer can be set to timeout in longer ranges of time intervals, typically in seconds.

The software watchdog timer can be enabled or disabled, depending on the operating condition. When enabled, system software must write to the Software Watchdog Timer register before it expires. When the timer does timeout, the system is preconfigured to generate an IRQ, an FIQ, or a RESET to restart the entire system.

Downloaded from www.Manualslib.com manuals search engine

General purpose timers/counters

Ten 32-bit general purpose timers/counters (GPTC) provide programmable time intervals to the CPU when used as one or multiple timers. There are two I/O pins associated with each timer.

- When used as a *gated* timer, one I/O pin serves as an input qualifier (high/low programmable).
- When used as a *regular* timer (enabled by software), the other I/O pin serves as a terminal count indicator output.

These pins can also be used independently as up/down counters to monitor the frequency of certain events (events capturing). In this situation, the I/O pin becomes the clock source of the counter.

Source clock frequency

Depending on the applications, the source clock frequency of the timers/counters is selectable among the system memory clock, the system memory clock with multiple divisor options, or an external pulse event. The divisor options are 2, 4, 6, 16, 32, 62, 128, or 256. If an external pulse is used, the frequency must be less than one half the system memory clock frequency.

GPTC characteristics

- Each GPTC can measure external event lengths up to minutes range, and can be individually enabled/disabled.
- Each GPTC can be configured to reload, with the value defined in the Initial Timer Count register (one for each GPTC), and generates an interrupt upon terminal count.
- Each GPTC has an interrupt request connected to the IRQ interrupt controller (VIC). The priority level and enable/disable of each interrupt can be programmed in the VIC. The CPU can read the contents of the timer/counter.
- GPTCs can be concatenated to form larger timer counters.

Control field

Include this control field in each of the 32-bit timer/counter control registers:

- Clock frequency selection
- Mode of operation:
  - Internal timer, with or without external terminal count indicator
  - External gated timer with gate active low
  - External gated timer with gate active high
  - External event counter — frequency must be less than one half the system memory clock frequency
- Timer/counter enable
- Count up or down

Downloaded from www.Manualslib.com manuals search engine

- Interrupt enable
- Concatenate to up-stream timer/counter; that is, use up-stream timer/counter's overflow/underflow output as clock input
- Reload enable
- Basic PWM function
- Enhanced PWM functionality (timers 6–9)
- Quadrature decoder function (timer 5)
- 32-bit or 16-bit operation

16-bit mode options

These options are available in 16-bit mode:

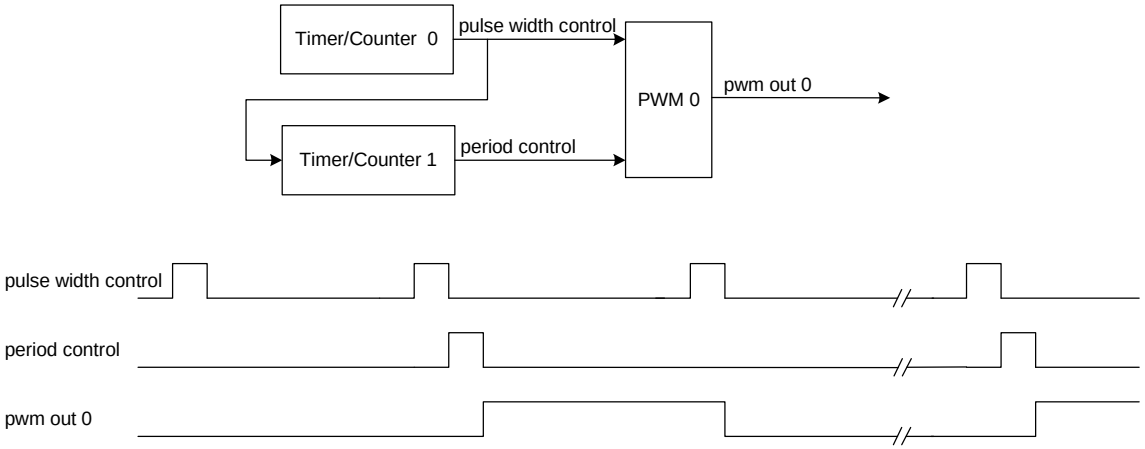
- **Capture mode.** Capture the counter value on the rising or falling edge of an external event and interrupt the CPU.
- **Compare mode.** Interrupt the CPU when the counter value is equal to the Match register.

Basic PWM function

Any of the timer/counters can be configured to provide a basic PWM function. Each PWM function requires concatenating two timer/counters, resulting in four PWM outputs. One of the timer/counters controls the pulse width and the other controls the period. The basic PWM function is output through GPIO through functions labeled PWM Ch *N*.

Functional block diagram

This diagram illustrates the basic PWM function:



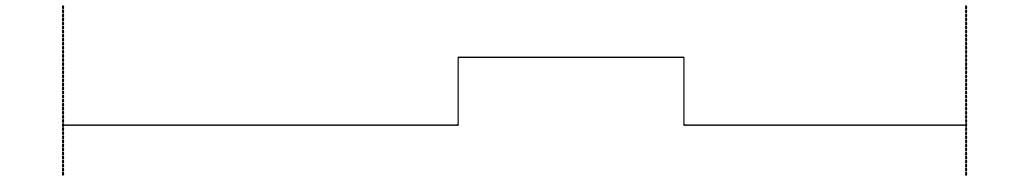
Downloaded from www.Manualslib.com manuals search engine

Enhanced PWM function

Timer counters 6–9 have additional features to add enhanced PWM functionality:

- High register — Compared to the timer/counter to toggle PWM output high
- Low register — Compared to the timer/counter to toggle PWM output back low
- Three 15-bit Step registers associated with four enhanced timer/counters. The values of Step registers are added when the high, low, and reload values are reached, which allows a steadily variable motor control PWM wave to be generated. The enhanced PWM function is output through GPIO through the functions labeled Ext Timer Event Out Ch N for channels 6 to 9.

Sample enhanced PWM waveform



Timer Counter clock frequency	75MHz
PWM period	873.000 usec
Low time 1	382.293 usec
High time	218.453 usec
Low time 2	272.254 usec

Quadrature decoder function

The processor provides a quadrature decoder function to allow the CPU to determine the external device rate of rotation and the direction of rotation. Example applications are robotic axles for feedback control, mechanical knobs to determine user input, and in computer mice, to determine direction of movement.

One timer/counter will include a quadrature decoder function, which takes some computational load off the CPU. When a CPU reads the output signals of a quadrature encoder, every state must be decoded and a counter needs to be updated based on the interpretation of the states. For example, for an encoder of 256 pulses per revolution turning at a modest 6000 rpm, the CPU needs to find and decoded 102,400 state changes per second and update the counter accordingly. With an x8 sampling rate, the CPU needs to sample the input about 8 x 102,400 timer per second. This consumes a significant portion of the CPU bandwidth.

Downloaded from www.Manualslib.com manuals search engine

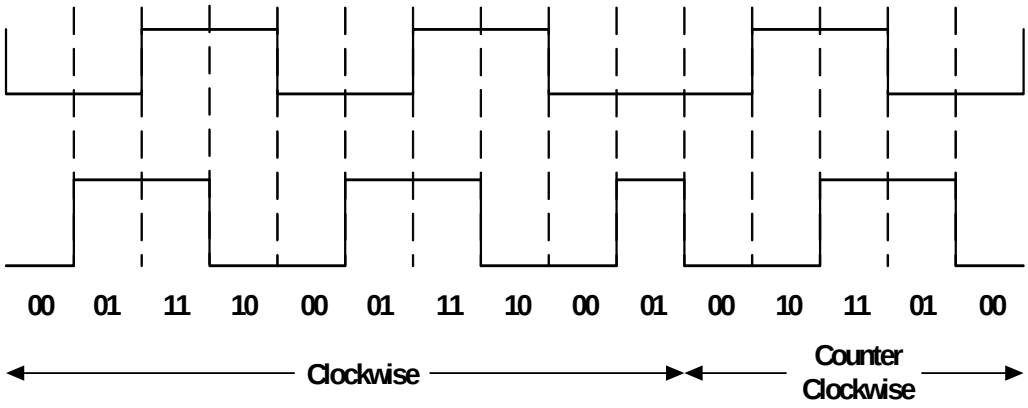
A quadrature decoder/counter module performs these tasks at real time speed and interrupts the CPU at the predetermined conditions.

How the quadrature decoder/counter works

Provides input signals

A quadrature encoder provides a pair of signals (in-phase and quad-phase) with opposite polarities and a 90-degree phase shift. Decode these signals to create an algorithm to determine the direction, speed, and position of a motion wheel.

Input signals



Quadrature encoding truth table

Legend:

NC — No change

CW — Clockwise

CCW — Counter clockwise

Err — Error

I:Q	00	01	10	11
00	NC	CW	CCW	Err
01	CCW	NC	Err	CW
10	CW	Err	NC	CCW
11	Err	CCW	CW	NC

Downloaded from www.Manualslib.com manuals search engine

Monitors how far the encoder has moved

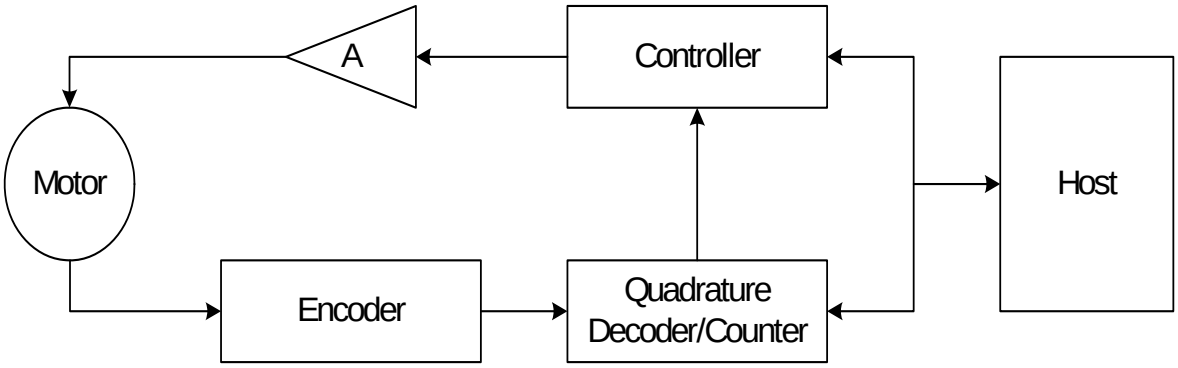
The counter keeps a running count of how far the encoder has moved.

- The decoder increments a 32-bit counter when a state change is found in the positive direction.
- The decoder decrements a 32-bit counter when a state change is found in the other direction.

When the programmed number reaches the terminal count, the counter is reset and an interrupt is generated to the CPU. The CPU can also read the counter directly to sense the direction of the motor.

Typical application

This diagram shows a typical application of the quadrature decoder/counter:



Digital filter

To ensure the precision and quality of the quadrature decoder/counter, a digital filter rejects noise on the incoming quadrature signals using three-clock-cycle delayed filtering. The three-clock-cycle delay filter rejects large and short duration noise spikes that typically occur in motor system applications.

Testing signals

Each signal is sampled on rising clock edges. A time history of the signals is stored in a four-bit shift register. Any signal is tested for a stable level that is present for three consecutive rising clock edges. With this method, pulses shorter than a two-clock period are rejected.

Timer support

Timer counter 5 supports the sampling clock and the counters.

Downloaded from www.Manualslib.com manuals search engine

Interrupt controller

The interrupt system is a simple two-tier priority scheme. Two lines access the CPU core and can interrupt the processor: IRQ (normal interrupt) and FIQ (fast interrupt). FIQ has a higher priority than IRQ.

FIQ interrupts

Most sources of interrupts on the processor are from the IRQ line. There is only one FIQ source for timing-critical applications. The FIQ interrupt generally is reserved for timing-critical applications for these reasons:

- The interrupt service routine is executed directly without determining the source of the interrupt.
- Interrupt latency is reduced. The banked registers available for FIQ interrupts are more efficient because a context save is not required.

Note: The interrupt source assigned to the FIQ must be assigned to the highest priority, which is 0.

IRQ interrupts

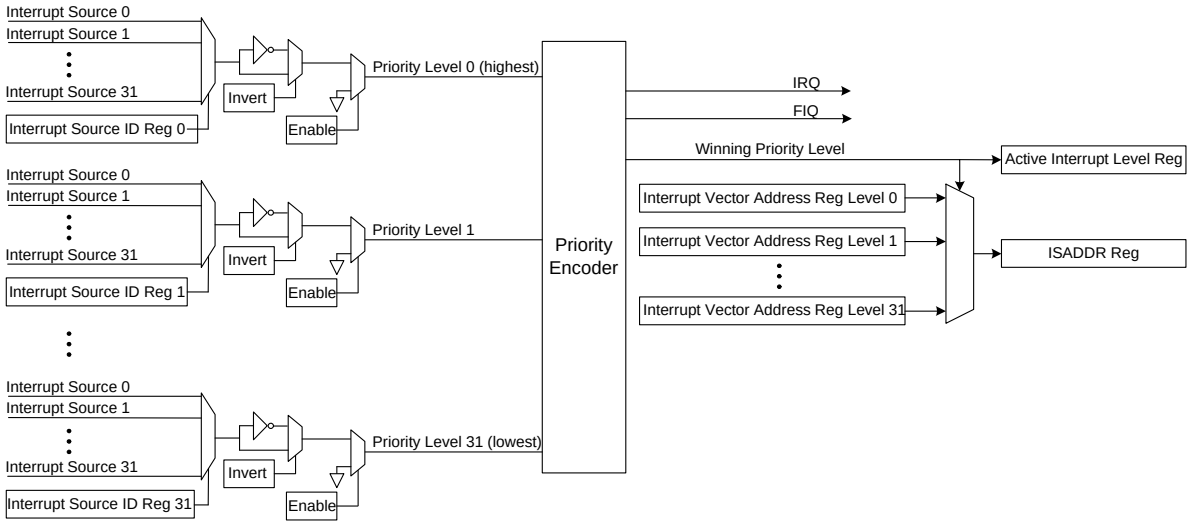
IRQ interrupts come from several different sources in the processor and are managed using the Interrupt Config registers (see “Int (Interrupt) Config (Configuration) 31–0 registers” on page 175). IRQ interrupts can be enabled or disabled on a per-level basis using the Interrupt Enable registers. These registers serve as masks for the different interrupt levels. Each interrupt level has two registers:

- **Interrupt Configuration register.** Use this register to assign the source for each interrupt level, invert the source polarity, select IRQ or FIQ, and enable the level.
- **Interrupt Vector Address register.** Contains the address of the interrupt service routine.

32-vector interrupt controller

The next figure shows a 32-vector interrupt controller:

Downloaded from www.Manualslib.com manuals search engine



IRQ characteristics

- The IRQ interrupts are enabled by the respective enabling bits.
- Once enabled, the interrupt source programmed in the Interrupt Configuration register for each priority level connects the interrupt to one of 32 priority lines going into the priority encoder block.
- The priority encoder block has a fixed order, with line 0 as the highest priority. The interrupt with the highest priority level has its encoded priority level displayed, to select the appropriate vector for the ISADDR register (see “ISADDR register” on page 176).
- The CPU, once interrupted, can read the ISADDR register to get the address of the Interrupt Service Routine. A read to the ISADDR register updates the priority encoder block, which masks the current and any lower priority interrupt requests. Writing to this address indicates to the priority hardware that the current interrupt is serviced, allowing lower priority interrupts to become active.
- The write value to the ISADDR register must be the level of the interrupt being serviced. Valid values are 0–31.
- The priority encoder block enables 32 prioritized interrupts to be serviced in nested fashion. A software interrupt can be implemented by writing to a software interrupt register. The software interrupt typically is assigned level 1 or level 2 priority.

Interrupt sources

An Interrupt Status register shows the current active interrupt requests. The Raw Interrupts register shows the status of the unmasked interrupt requests.

Downloaded from www.Manualslib.com manuals search engine

The interrupt sources are assigned as shown:

Interrupt ID	Interrupt source
0	Watchdog Timer
1	AHB Bus Error
2	Ext DMA
3	CPU Wake Interrupt
4	Ethernet Module Receive Interrupt
5	Ethernet Module Transmit Interrupt
6	Ethernet Phy Interrupt
7	UART A Interrupt
8	UART B Interrupt
9	UART C Interrupt
10	UART D Interrupt
11	SPI Interrupt
12	Reserved
13	Reserved
14	ADC Interrupt
15	Early Power Loss Interrupt
16	I^2C Interrupt
17	RTC Interrupt
18	Timer Interrupt 0
19	Timer Interrupt 1
20	Timer Interrupt 2
21	Timer Interrupt 3
22	Timer Interrupt 4
23	Timer Interrupt 5
24	Timer Interrupt 6
25	Timer Interrupt 7
26	Timer Interrupt 8
27	Timer Interrupt 9
28	External Interrupt 0
29	External Interrupt 1
30	External Interrupt 2
31	External Interrupt 3

Downloaded from www.Manualslib.com manuals search engine

Vectored interrupt controller (VIC) flow

This is how the VIC flow works:

1. An interrupt occurs.
2. The CPU branches to either the IRQ or FIQ interrupt vector.
3. If the CPU goes to the IRQ vector, the CPU reads the service routine address from the VIC's ISADDR register. The READ updates the VIC's priority hardware to prevent current or any lower priority interrupts from interrupting again. The CPU must not read the ISADDR register for FIQ interrupts.
4. The CPU branches to the Interrupt Service Routine (ISR) and stacks the workspace so the IRQ can be enabled.
5. The CPU enables the IRQ interrupts so higher priority interrupts can be serviced.
6. The CPU executes the interrupt service routine.
7. The CPU clears the source of the current interrupt.
8. The CPU disables the IRQ and restores the workspace.
9. If IRQ, the CPU writes the level value of the interrupt being serviced to the ISADDR register to clear the current interrupt path in the VIC's priority hardware.
10. The CPU returns from the interrupt.

Configurable system attributes

System software can configure these system attributes:

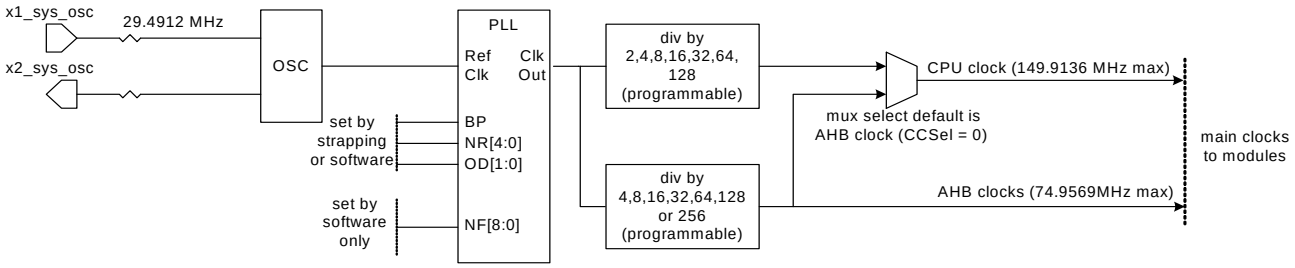
- Little endian/big endian mode
- Watchdog timer enable
- Watchdog timeout generates IRQ/FIQ/RESET
- Watchdog timeout interval
- Enable/disable ERROR response for misaligned data access
- System module clock enables
- Enable access to internal registers in USER mode

PLL configuration

Hardware strapping determines the initial powerup PLL (see "Bootstrap initialization" on page 152). After powerup, software can change the PLL settings by writing to the PLL Configuration register.

Downloaded from www.Manualslib.com manuals search engine

PLL configuration and control system block diagram



PLL Vco = (RefClk / NR+1) * (NF+1)
ClkOut = PLL Vco / OD+1

defaults

NR + 1 = 8
OD + 1 = 1
NF + 1 = 61

Sample Clock Frequency Settings With 29.4912MHz Crystal
(NF+1 = 61 and OD + 1 = 1)

NR+1	Clk Out	CPU clock (CCSel = 1)	CPU clock (CCSel = 0)	AHB clock
6	299.8272	149.9136	74.9568	74.9568
7	256.9947	128.4975	64.2487	64.2487
8	**224.8704**	112.4352	**56.2176**	**56.2176**
9	199.8848	99.9424	49.9712	49.9712
10	179.8962	89.9482	44.9741	44.9741
11	163.5421	81.7711	40.8855	40.8855
12	149.9136	74.9568	37.4784	37.4784

Restrictions
RefClk / NR+1 range: 275KHz – 550MHz
PLL Vco range: 110MHz – 550MHz

Bootstrap initialization

The PLL and other system configuration settings can be configured at powerup before the CPU boots. External pins configure the necessary control register bits at powerup. There are internal pullup resistors on these pins to provide a default configuration. External pulldown resistors can configure the PLL and system configuration registers depending on the application.

Configuring the powerup settings

This table shows how each bit configures the powerup settings.

- 0 = Use an external pulldown
- 1 = Use the internal pullup

Downloaded from www.Manualslib.com manuals search engine

Pin name	Configuration bits
gpio_a[3]	Endian configuration 0 Little endian 1 Big endian
gpio_a[2]	Boot mode 0 Boot from SDRAM using serial SPI EEPROM 1 Boot from Flash ROM
gpio_a[0], addr[23]	Flash/SPI configuration If booting from Flash: 00 8 bit 01 32 bit 10 32 bit 11 16 bit If booting from SPI 00 Reserved 01 8-bit addressing 10 24-bit addressing 11 16-bit addressing
addr[19:9]	Gen ID
addr[7]	PLL bypass setting 0 Bypass 1 Normal operation
addr[6:5]	PLL output divider setting OD 00 3 01 2 10 1 11 0
addr[4:0]	PLL reference clock divider setting NR **00111** 31 **01100** 20 **10001** 9 **00110** 30 **01011** 19 **10000** 8 **00101** 29 **01010** 18 **11111** 7 **00100** 28 **01001** 17 **11110** 6 **00011** 27 **01000** 16 **11101** 5 **00010** 26 **10111** 15 **11100** 4 **00001** 25 **10110** 14 **11011** 3 **00000** 24 **10101** 13 **11010** 2 **01111** 23 **10100** 12 **11001** 1 **01110** 22 **10011** 11 **11000** 0 **01101** 21 **10010** 10

Downloaded from www.Manualslib.com manuals search engine

System configuration registers

All configuration registers must be accessed as 32-bit words and as single accesses only. Bursting is not allowed.

Register address map

Offset	[31:24]	[23:16]	[15:8]	[7:0]
A090 0000	General Arbiter Control			
A090 0004	BRC0			
A090 0008	BRC1			
A090 000C	BRC2			
A090 0010	BRC3			
A090 0014	Reserved			
A090 0018	AHB Error Detect Status 1			
A090 001C	AHB Error Detect Status 2			
A090 0020	AHB Error Monitoring Configuration			
A090 0024	Timer Master Control			
A090 0028	Timer 0 Reload Count and Compare register			
A090 002C	Timer 1 Reload Count and Compare register			
A090 0030	Timer 2 Reload Count and Compare register			
A090 0034	Timer 3 Reload Count and Compare register			
A090 0038	Timer 4 Reload Count and Compare register			
A090 003C	Timer 5 Reload Count and Compare register			
A090 0040	Timer 6 Reload Count and Compare register			
A090 0044	Timer 7 Reload Count and Compare register			
A090 0048	Timer 8 Reload Count and Compare register			
A090 004C	Timer 9 Reload Count and Compare register			
A090 0050	Timer 0 Read and Capture register			
A090 0054	Timer 1 Read and Capture register			
A090 0058	Timer 2 Read and Capture register			
A090 005C	Timer 3 Read and Capture register			
A090 0060	Timer 4 Read and Capture register			
A090 0064	Timer 5 Read and Capture register			
A090 0068	Timer 6 Read and Capture register			
A090 006C	Timer 7 Read and Capture register			
A090 0070	Timer 8 Read and Capture register			

Downloaded from www.Manualslib.com manuals search engine

Offset	[31:24]	[23:16]	[15:8]	[7:0]
A090 0074	Timer 9 Read and Capture register			
A090 0078	Timer 6 High register			
A090 007C	Timer 7 High register			
A090 0080	Timer 8 High register			
A090 0084	Timer 9 High register			
A090 0088	Timer 6 Low register			
A090 008C	Timer 7 Low register			
A090 0090	Timer 8 Low register			
A090 0094	Timer 9 Low register			
A090 0098	Timer 6 High and Low Step register			
A090 009C	Timer 7 High and Low Step register			
A090 00A0	Timer 8 High and Low Step register			
A090 00A4	Timer 9 High and Low Step register			
A090 00A8	Timer 6 Reload Step register			
A090 00AC	Timer 7 Reload Step register			
A090 00B0	Timer 8 Reload Step register			
A090 00B4	Timer 9 Reload Step register			
A090 00B8	Reserved			
A090 00BC	Reserved			
A090 00C0	Reserved			
A090 00C4	Interrupt Vector Address Register Level 0			
A090 00C8	Interrupt Vector Address Register Level 1			
A090 00CC	Interrupt Vector Address Register Level 2			
A090 00D0	Interrupt Vector Address Register Level 3			
A090 00D4	Interrupt Vector Address Register Level 4			
A090 00D8	Interrupt Vector Address Register Level 5			
A090 00DC	Interrupt Vector Address Register Level 6			
A090 00E0	Interrupt Vector Address Register Level 7			
A090 00E4	Interrupt Vector Address Register Level 8			
A090 00E8	Interrupt Vector Address Register Level 9			
A090 00EC	Interrupt Vector Address Register Level 10			
A090 00F0	Interrupt Vector Address Register Level 11			
A090 00F4	Interrupt Vector Address Register Level 12			

Downloaded from www.Manualslib.com manuals search engine

Offset	[31:24]	[23:16]	[15:8]	[7:0]
A090 00F8	Interrupt Vector Address Register Level 13			
A090 00FC	Interrupt Vector Address Register Level 14			
A090 0100	Interrupt Vector Address Register Level 15			
A090 0104	Interrupt Vector Address Register Level 16			
A090 0108	Interrupt Vector Address Register Level 17			
A090 010C	Interrupt Vector Address Register Level 18			
A090 0110	Interrupt Vector Address Register Level 19			
A090 0114	Interrupt Vector Address Register Level 20			
A090 0118	Interrupt Vector Address Register Level 21			
A090 011C	Interrupt Vector Address Register Level 22			
A090 0120	Interrupt Vector Address Register Level 23			
A090 0124	Interrupt Vector Address Register Level 24			
A090 0128	Interrupt Vector Address Register Level 25			
A090 012C	Interrupt Vector Address Register Level 26			
A090 0130	Interrupt Vector Address Register Level 27			
A090 0134	Interrupt Vector Address Register Level 28			
A090 0138	Interrupt Vector Address Register Level 29			
A090 013C	Interrupt Vector Address Register Level 30			
A090 0140	Interrupt Vector Address Register Level 31			
A090 0144	Int Config 0	Int Config 1	Int Config 2	Int Config 3
A090 0148	Int Config 4	Int Config 5	Int Config 6	Int Config 7
A090 014C	Int Config 8	Int Config 9	Int Config 10	Int Config 11
A090 0150	Int Config 12	Int Config 13	Int Config 14	Int Config 15
A090 0154	Int Config 16	Int Config 17	Int Config 18	Int Config 19
A090 0158	Int Config 20	Int Config 21	Int Config 22	Int Config 23
A090 015C	Int Config 24	Int Config 25	Int Config 26	Int Config 27
A090 0160	Int Config 28	Int Config 29	Int Config 30	Int Config 31
A090 0164	ISADDR			
A090 0168	Interrupt Status Active			
A090 016C	Interrupt Status Raw			
A090 0170	Reserved			
A090 0174	Software Watchdog Configuration			
A090 0178	Software Watchdog Timer			

Downloaded from www.Manualslib.com manuals search engine

Offset	[31:24] [23:16] [15:8] [7:0]
A090 017C	Clock Configuration register
A090 0180	Module Reset register
A090 0184	Miscellaneous System Configuration register
A090 0188	PLL Configuration register
A090 018C	Active Interrupt ID register
A090 0190	Timer 0 Control register
A090 0194	Timer 1 Control register
A090 0198	Timer 2 Control register
A090 019C	Timer 3 Control register
A090 01A0	Timer 4 Control register
A090 01A4	Timer 5 Control register
A090 01A8	Timer 6 Control register
A090 01AC	Timer 7 Control register
A090 01B0	TImer 8 Control register
A090 01B4	Timer 9 Control register
A090 01B8 – A090 01CC	Reserved
A090 01D0	System Memory Chip Select 0 Dynamic Memory Base
A090 01D4	System Memory Chip Select 0 Dynamic Memory Mask
A090 01D8	System Memory Chip Select 1 Dynamic Memory Base
A090 01DC	System Memory Chip Select 1 Dynamic Memory Mask
A090 01E0	System Memory Chip Select 2 Dynamic Memory Base
A090 01E4	System Memory Chip Select 2 Dynamic Memory Mask
A090 01E8	System Memory Chip Select 3 Dynamic Memory Base
A090 01EC	System Memory Chip Select 3 Dynamic Memory Mask
A090 01F0	System Memory Chip Select 0 Static Memory Base
A090 01F4	System Memory Chip Select 0 Static Memory Mask
A090 01F8	System Memory Chip Select 1 Static Memory Base
A090 01FC	System Memory Chip Select 1 Static Memory Mask
A090 0200	System Memory Chip Select 2 Static Memory Base
A090 0204	System Memory Chip Select 2 Static Memory Mask
A090 0208	System Memory Chip Select 3 Static Memory Base
A090 020C	System Memory Chip Select 3 Static Memory Mask
A090 0210	Gen ID

Downloaded from www.Manualslib.com manuals search engine

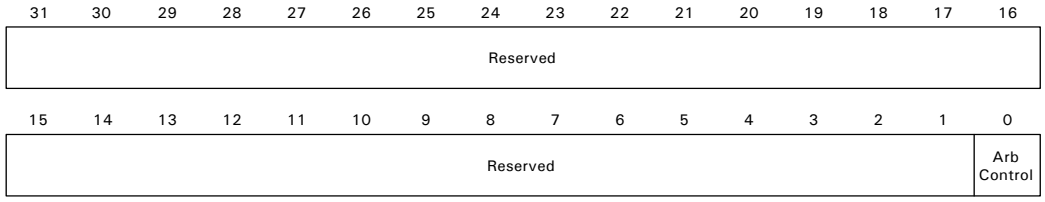
Offset	[31:24]	[23:16]	[15:8]	[7:0]
A090 0214	External Interrupt 0 Control register			
A090 0218	External Interrupt 1 Control register			
A090 021C	External Interrupt 2 Control register			
A090 0220	External Interrupt 3 Control register			
A090 0224	RTC Module Control			
A090 0228	Power Management			
A090 022C	AHB Bus Activity Status			

General Arbiter Control register

Address: A090 0000

The General Arbiter Control register controls whether the CPU access is routed through the main arbiter or is connected directly to the memory controller.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:01	N/A	Reserved	N/A	N/A
D00	R	ArbControl	0x0	**Arbiter control** 0 CPU connected directly to memory controller 1 CPU connected to main arbiter

BRC0, BRC1, BRC2, and BRC3 registers

Addresses: A090 0004 / 0008 / 000C / 0010

The BRC[0:3] registers control the AHB arbiter bandwidth allocation scheme.

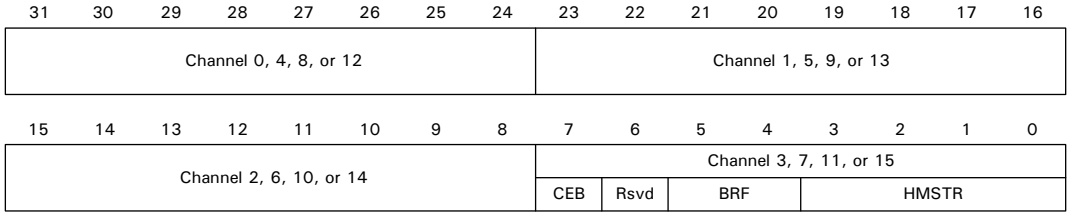
Downloaded from www.Manualslib.com manuals search engine

Channel allocation

This is how the channels are assigned in the four registers:

Register name	[31:24]	[23:16]	[15:08]	[07:00]
BRC0	Channel 0	Channel 1	Channel 2	Channel 3
BRC1	Channel 4	Channel 5	Channel 6	Channel 7
BRC2	Channel 8	Channel 9	Channel 10	Channel 11
BRC3	Channel 12	Channel 13	Channel 14	Channel 15

Register



Register bit assignment

This table shows the bit definition for each channel, using data bits [07:00] as the example.

Bits	Access	Mnemonic	Reset	Description
D07	R/W	CEB	0x0	**Channel enable bit** 0 Disabled 1 Enabled
D06	N/A	Reserved	N/A	**N/A**
D05:04	R/W	BRF	0x0	Bandwidth reduction field Program the weight for each AHB bus master. Used to limit access to the round robin scheduler. 00 100% 01 75% 10 50% 11 25%
D03:00	R/W	HMSTR	0x0	hmaster Program a particular AHB bus master number here. Note that a particular master an be programmed to more than one channel.

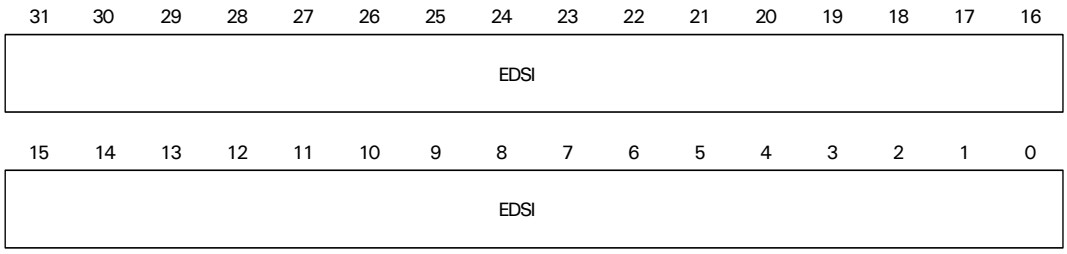
AHB Error Detect Status 1

Address: A090 0018

Downloaded from www.Manualslib.com manuals search engine

The AHB Error Detect Status 1 register records the haddr[31:0] value present when any AHB error is found. Note that this value is not reset on powerup but is reset when the AHB Error Interrupt Clear bit is set in the AHB Error Monitoring Configuration register (*).

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:00	*	EDS1	Not reset	The haddr[31:0] value recorded during a slave error response.

AHB Error Detect Status 2

Address: A090 001C

The AHB Error Detect Status 2 register records AHB master and slave values present when any AHB error is found. This register also records which error condition was triggered. Note that this value is not reset on powerup but is reset when the AHB Interrupt Clear bit is set in the AHB Error Monitoring Configuration register (*).

Register

Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:20	N/A	Reserved	N/A	**N/A**
D19	*	IE	Not reset	**CPU instruction error** An error was found on the CPU instruction access to external memory. The other fields in this register and the AHB Error Status 1 register are not valid if this bit is set.
D18	*	DE	Not reset	**CPU data error** An error was found on the CPU data access to external memory. The other fields in this register and the AHB Error Status 1 register are not valid if this field is set.
D17	*	ER	Not reset	**AHB error response** Set if an AHB slave ERROR response is found.
D16:15	N/A	Reserved	N/A	N/A
D14	*	HWR	Not reset	hwrite **Transaction type: write or read.**
D13:10	*	HMSTR	Not reset	**hmaster[3:0]** Initiating master identifier.
D09:06	*	HPR	Not reset	**hprot[3:0]** Transaction protection code.
D05:03	*	HSZ	Not reset	**hsize[2:0]** Transaction size.
D02:00	*	HBRST	Not reset	hburst[2:0 Transaction burst type

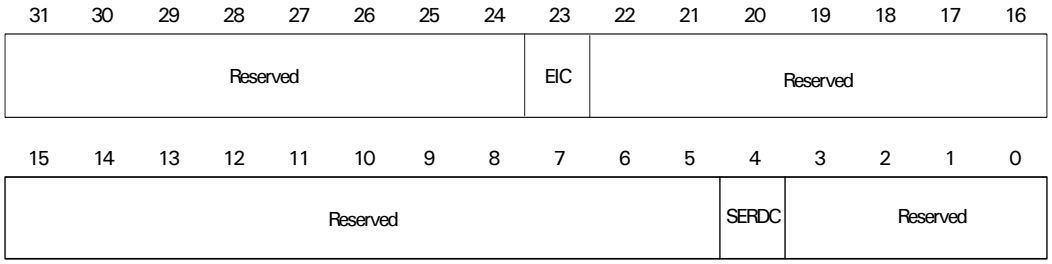
AHB Error Monitoring Configuration register

Address: A090 0020

The AHB Error Monitoring Configuration register configures the AHB arbiter error monitoring settings.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

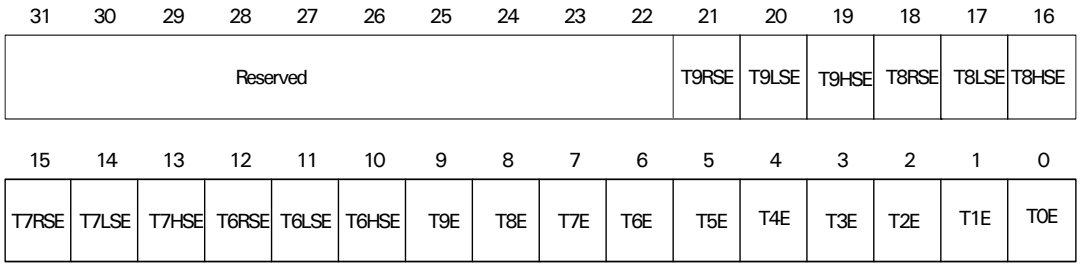
Bits	Access	Mnemonic	Reset	Description
D31:24	N/A	Reserved	N/A	N/A
D23	R/W	EIC	0x0	**AHB Error Interrupt Clear** Write a 1, then a 0 to this register to clear the AHB error interrupt and to clear the AHB Error Detect Status 1 and AHB Error Detect Status 2 registers.
D22:05	N/A	Reserved	N/A	N/A
D04	R/W	SERDC	0x0	**AHB Slave Error Response Detect Config** 0 Record error only 1 Generate IRQ
D03:00	N/A	Reserved	N/A	**N/A**

Timer Master Control register

Address: A090 0024

The Timer Master Control register resets and enables the timer in groups, which is useful when using the timers in PW applications.

Register



Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:22	N/A	Reserved	N/A	N/A
D21	R/W	T9RSE	0x0	**Timer 9 reload step enable** 0 Reload Step register disabled 1 Reload Step register enabled
D20	R/W	T9LSE	0x0	**Timer 9 low step enable** 0 Low Step register disabled 1 Low Step register enabled
D19	R/W	T9HSE	0x0	**Timer 9 high step enable** 0 High Step register disabled 1 High Step register enabled
D18	R/W	T8RSE	0x0	**Timer 8 reload step enable** 0 Reload Step register disabled 1 Reload Step register enabled
D17	R/W	T8LSE	0x0	**Timer 8 low step enable** 0 Low Step register disabled 1 Low Step register enabled
D16	R/W	T8HSE	0x0	**Timer 8 high step enable** 0 High Step register disabled 1 High Step register enabled
D15	R/W	T7RSE	0x0	**Timer 7 reload step enable** 0 Reload Step register disabled 1 Reload Step register enabled
D14	R/W	T7LSE	0x0	**Timer 7 low step enable** 0 Low Step register disabled 1 Low Step register enabled
D13	R/W	T7HSE	0x0	**Timer 7 high step enable** 0 High Step register disabled 1 High Step register enabled
D12	R/W	T6RSE	0x0	**Timer 6 reload step enable** 0 Reload Step register disabled 1 Reload Step register enabled
D11	R/W	T6LSE	0x0	**Timer 6 low step enable** 0 Low Step register disabled 1 Low Step register enabled
D10	R/W	T6HSE	0x0	**Timer 6 high step enable** 0 High Step register disabled 1 High Step register enabled

Downloaded from www.Manualslib.com manuals search engine

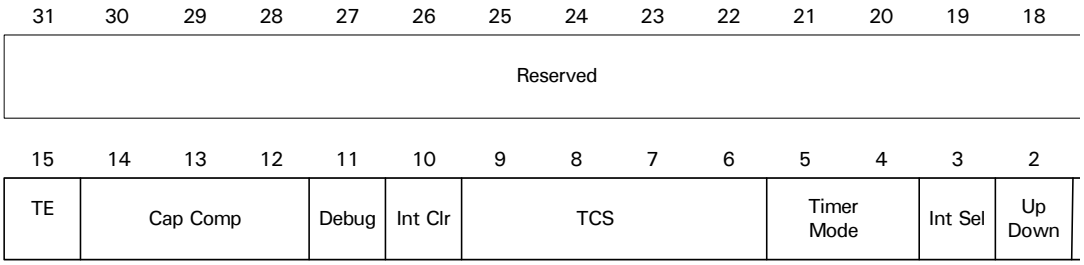
Bits	Access	Mnemonic	Reset	Description
D09	R/W	T9E	0x0	**Timer 9 enable** 0 Timer reset 1 Timer enabled
D08	R/W	T8E	0x0	**Timer 8 enable** 0 Timer reset 1 Timer enabled
D07	R/W	T7E	0x0	**Timer 7 enable** 0 Timer reset 1 Timer enabled
D06	R/W	T6E	0x0	**Timer 6 enable** 0 Timer reset 1 Timer enabled
D05	R/W	T5E	0x0	**Timer 5 enable** 0 Timer reset 1 Timer enabled
D04	R/W	T4E	0x0	**Timer 4 enable** 0 Timer reset 1 Timer enabled
D03	R/W	T3E	0x0	**Timer 3 enable** 0 Timer reset 1 Timer enabled
D02	R/W	T2E	0x0	**Timer 2 enable** 0 Timer reset 1 Timer enabled
D01	R/W	T1E	0x0	**Timer 1 enable** 0 Timer reset 1 Timer enabled
D00	R/W	T0E	0x0	**Timer 0 enable** 0 Timer reset 1 Timer enabled

Timer 0-4 Control registers

Addresses: A090 0190 / 0194 / 0198 / 019C / 01A0

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:16	N/A	Reserved	N/A	N/A
D15	R/W	TE	0x0	**Timer enable** 0 Timer is disabled 1 Timer is enabled
D14:12	R/W	Cap Comp	0x0	**Capture and compare mode functions** Applicable only when in 16-bit timer mode. 000 Normal operation 001 Compare mode, toggle output on match 010 Compare mode, pulse output on match 011 Capture mode, on input falling edge 100 Capture mode, on input rising edge 101 Capture mode, on every 2nd rising edge 110 Capture mode, on every 4th rising edge 111 Capture mode, on every 8th rising edge
D11	R/W	Debug	0x0	**Debug mode** 0 Timer enabled in CPU debug mode 1 Timer disabled in CPU debug mode
D10	R/W	Int Clr	0x0	**Interrupt clear** Clears the timer interrupt. Software must write a 1, then a 0 to this location to clear the interrupt.
D09:06	R/W	TCS	0x0	**Timer clock select** 0000 AHB clock x 2 0001 AHB clock 0010 AHB clock / 2 0011 AHB clock / 4 0100 AHB clock / 8 0101 AHB clock / 16 0110 AHB clock / 32 0111 AHB clock / 64 1000 AHB clock / 128 1111 External event

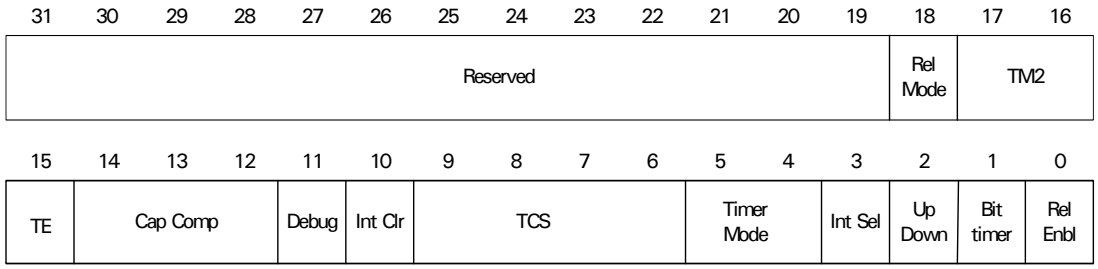
Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Reset	Description
D05:04	R/W	Timer mode	0x0	**Timer mode** 00 Internal timer or external event 01 External low-level gated timer 10 External high-level gated timer 11 Concatenate the lower timer. Note: When either external gated option is selected, the time clock select bits determine the frequency.
D03	R/W	Int Sel	0x0	**Interrupt select** 0 Interrupt disable 1 Generate IRQ
D02	R/W	Up Down	0x0	**Up/Down select** 0 Up counter 1 Down counter
D01	R/W	Bit timer	0x0	**32 or 16 bit timer** 0 16-bit timer 1 32-bit timer
D00	R/W	Rel Enbl	0x0	**Reload enable** 0 Halt at terminal count. The timer must be disabled, then enabled to reload the timer when the terminal count is reached. 1 Reload and resume count at terminal count

Timer 5 Control register

Address: A090 01A4

Register



Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:19	N/A	Reserved	N/A	N/A
D18	R/W	Rel mode	0x0	**Reload mode** Initializes the timer and the reload value at terminal count. Reload mode is useful in quadrature decoder applications, as it allows the reload value to be half of he terminal count. 0 Use the value in the Reload register 1 Use half the value in the Reload register
D17:16	R/W	TM2	0x0	**Timer mode 2** 00 Mode as set by timer mode 1 01 Reserved 10 Reserved 11 Quadrature decoder/counter mode
D15	R/W	TE	0x0	**Timer enable** 0 Timer disabled 1 Timer enabled
D14:12	R/W	Cap Comp	0x0	**Capture and compare mode functions** Applicable only when in 16-bit timer mode. 000 Normal operation 001 Compare mode, toggle output on match 010 Compare mode, pulse output on match 011 Capture mode, on input falling edge 100 Capture mode, on input rising edge 101 Capture mode, on every 2nd rising edge 110 Capture mode, on every 4th rising edge 111 Capture mode, on every 8th rising edge
D11	R/W	Debug	0x0	**Debug mode** 0 Timer enabled in CPU debug mode 1 Timer disabled in CPU debug mode
D10	R/W	Int Clr	0x0	**Interrupt clear** Clears the timer interrupt. Software must write a 1, then a 0 to this location to clear the interrupt.

Downloaded from www.Manualslib.com manuals search engine

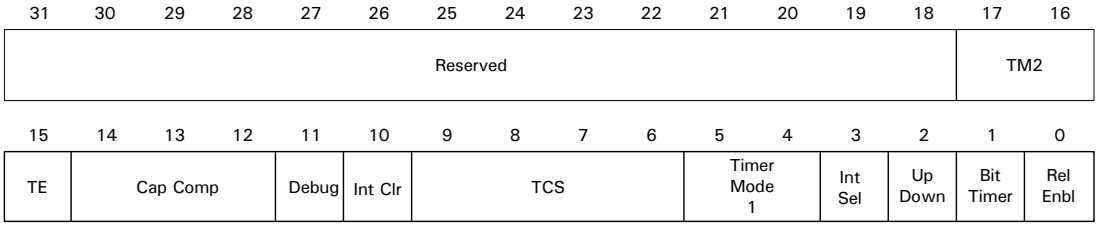
Bits	Access	Mnemonic	Reset	Description
D09:06	R/W	TCS	0x0	**Timer clock select** 0000 AHB clock x 2 0001 AHB clock 0010 AHB clock / 2 0011 AHB clock / 4 0100 AHB clock / 8 0101 AHB clock / 16 0110 AHB clock / 32 0111 AHB clock / 64 1000 AHB clock / 128 1111 External event
D05:04	R/W	Timer mode 1	0x0	**Timer mode 1** 00 Internal timer or external event 01 External low-level gated timer 10 External high-level gated timer 11 Concatenate the lower timer. Note: When either external gated option is selected, the time clock select bits determine the frequency.
D03	R/W	Int Sel	0x0	**Interrupt select** 0 Interrupt disable 1 Generate IRQ
D02	R/W	Up Down	0x0	**Up/Down select** 0 Up counter 1 Down counter
D01	R/W	Bit timer	0x0	**32 or 16 bit timer** 0 16-bit timer 1 32-bit timer
D00	R/W	Rel Enbl	0x0	**Reload enable** 0 Halt at terminal count. The timer must be disabled, then enabled to reload the timer when the terminal count is reached. 1 Reload and resume count at terminal count

Timer 6-9 Control registers

Addresses: A090 01A8 / 01AC / 01B0 / 01B4

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:18	N/A	Reserved	N/A	N/A
D17:16	R/W	TM2	0x0	**Timer mode 2** 00 Mode as set by timer mode 1 01 PWM mode, using High, Low, and Step registers 10 Clock mode, toggle the timer output at the terminal count to create a clock output 11 Reserved
D15	R/W	TE	0x0	**Timer enable** 0 Timer disabled 1 Timer enabled
D14:12	R/W	Cap Comp	0x0	**Capture and compare mode functions** Applicable only when in 16-bit timer mode. 000 Normal operation 001 Compare mode, toggle output on match 010 Compare mode, pulse output on match 011 Capture mode, on input falling edge 100 Capture mode, on input rising edge 101 Capture mode, on every 2nd rising edge 110 Capture mode, on every 4th rising edge 111 Capture mode, on every 8th rising edge
D11	R/W	Debug	0x0	**Debug mode** 0 Timer enabled in CPU debug mode 1 Timer disabled in CPU debug mode
D10	R/W	Int Clr	0x0	**Interrupt clear** Clears the timer interrupt. Software must write a 1, then a 0 to this location to clear the interrupt.

Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Reset	Description
D09:06	R/W	TCS	0x0	**Timer clock select** 0000 AHB clock x 2 (Not applicable if timer mode 2 is set to PWM mode (01)) 0001 AHB clock 0010 AHB clock / 2 0011 AHB clock / 4 0100 AHB clock / 8 0101 AHB clock / 16 0110 AHB clock / 32 0111 AHB clock / 64 1000 AHB clock / 128 1111 External event
D05:04	R/W	Timer mode 1	0x0	**Timer mode 1** 00 Internal timer or external event 01 External low-level gated timer 10 External high-level gated timer 11 Concatenate the lower timer. When either external gated option is selected, the time clock select bits determine the frequency.
D03	R/W	Int Sel	0x0	**Interrupt select** 0 Interrupt disable 1 Generate IRQ
D02	R/W	Up Down	0x0	**Up/Down select** 0 Up counter 1 Down counter
D01	R/W	Bit timer	0x0	**32 or 16 bit timer** 0 16-bit timer 1 32-bit timer
D00	R/W	Rel Enbl	0x0	**Reload enable** 0 Halt at terminal count. The timer must be disabled, then enabled to reload the timer when the terminal count is reached. 1 Reload and resume count at terminal count

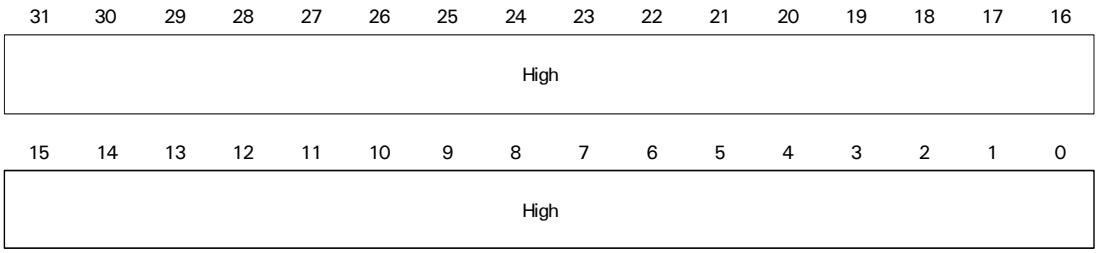
Timer 6–9 High registers

Addresses: A090 0078 / 007C / 0080 / 0084

The Timer 6–9 High registers contains the high registers for the enhanced PWM features available in timers 6 through 9.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

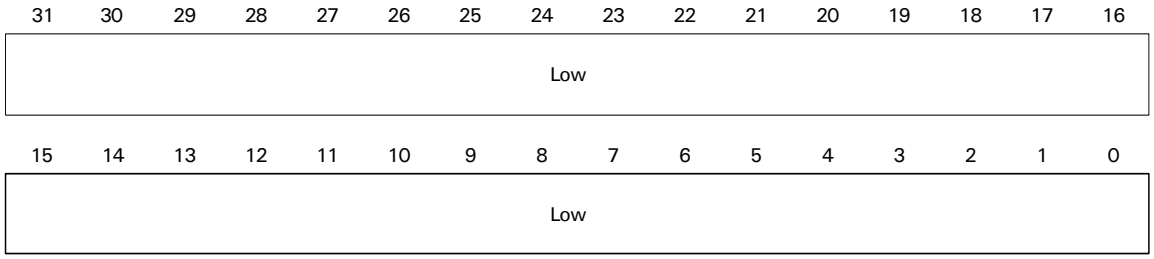
Bits	Access	Mnemonic	Reset	Description
D31:00	R/W	High	0x0	The PWM output toggles high when the timer counter reaches this value.

Timer 6-9 Low registers

Addresses: A090 0088 / 008C / 0090 / 0094

The Timer 6–9 Low registers contain the low registers for the enhanced PWM features available in timers 6 through 9.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:00	R/W	Low	0x0	The PWM output toggles low when the timer counter reaches this value.

Downloaded from www.Manualslib.com manuals search engine

Timer 6–9 High and Low Step registers

Addresses: A090 0098 / 009C / 00A0 / 00A4

The Timer 6–9 High and Low Step registers contain the high and low step registers for the enhanced PWM features available in timers 6 through 9.

Register

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31	R/W	Hi Step Dir	0x0	**High step direction** 0 Subtract the high step value from the original high register value to increase the high time. 1 Add the high step value to the original high register value to decrease the high time.
D30:16	R/W	Hi Step	0x0	**High step** This value is either added or subtracted from the original high register value once each cycle.
D15	R/W	Lo Step Dir	0x0	**Low step direction** 0 Subtract the low step value from the original low register value to increase low time 2. 1 Add the low step value to the original low register value to decrease low time 2.
D14:00	R/W	Lo Step	0x0	Low step This value is either added or subtracted from the original low register value once each cycle.

Timer 6–9 Reload Step registers

Addresses: A090 00A8 / 00AC / 00B0 / 00B4

The Timer 6–9 reload Step registers contain the reload step registers for the enhanced PWM features available in timers 6 through 9.

Downloaded from www.Manualslib.com manuals search engine

Register

Register bit assignment

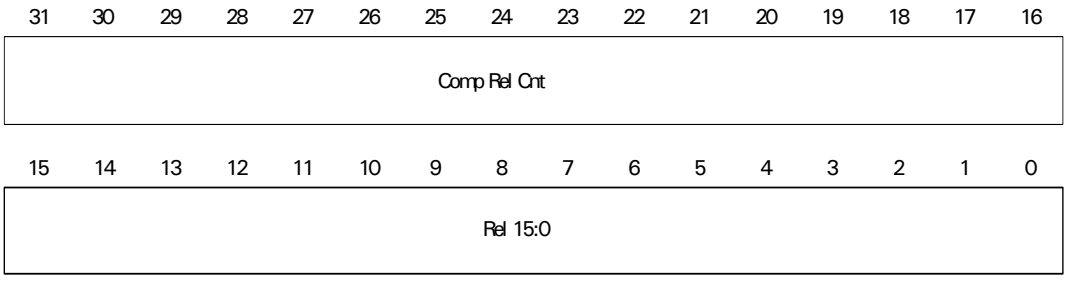
Bits	Access	Mnemonic	Reset	Description
D31:16	N/A	Reserved	N/A	N/A
D15	R/W	Rel Dir	0x0	**Reload step direction** 0 Subtract the reload step value from the original reload register value to increase the overall period. 1 Add the reload step value to the original reload register value to decrease the overall period.
D14:00	R/W	Rel Step	0x0	**Reload step** This value is either added or subtracted from the original low register value once each cycle.

Timer 0-9 Reload Count and Compare register

Addresses: A090 0028 / 002C / 0030 / 0034 / 0038 / 003C / 0040 / 0044 / 0048 / 004C

The Timer 0 to 9 Reload Count and Compare register holds the up/down reload and compare values for timers 0 to 9.

Register



Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

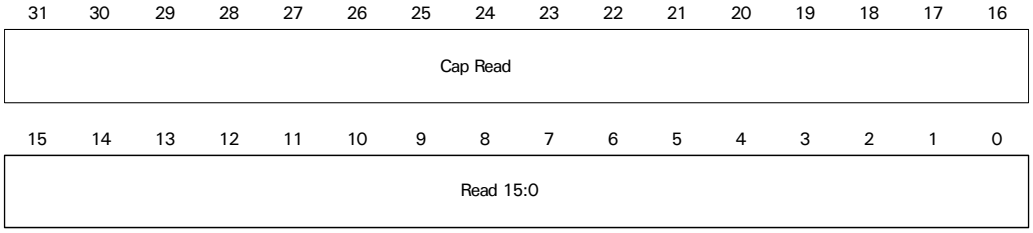
Bits	Access	Mnemonic	Reset	Description
D31:16	R/W	Comp Rel Cnt	0x0	**Timer Compare register or Timer Reload Bits 31:16 Count register** An external toggle or pulse is generated each time the timer value matches this value. An interrupt is generated, if enabled. If configured for a 32-bit timer, bits 31:16 timer reload.
D15:00	R/W	Rel 15:0	0x0	**Timer Reload Bits 15:00 Count register** This value is loaded into the Timer register after the timer is enable and after the terminal count has been reached if the reload enable bit is set.

Timer 0-9 Read and Capture register

Addresses: A090 0050 / 0054 / 0058 / 005C / 0060 / 0064 / 0068 / 006C / 0070 / 0074

The Timer 0 to 9 Read and Capture register reads the current state of each timer and capture register.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:16	R/W	Cap Read	0x0	**Timer Capture register or Timer Read Bits 31:16 register** Reads the capture value of each timer. An interrupt is generated on a capture event, if enabled. If configured as a 32-bit timer, then bits 31:16 of the current state of each timer.
D15:00	R/W	Read 15:0	0x0	**Timer Read Bits 15:00 register** Reads bits 15:00 of the current state of each timer.

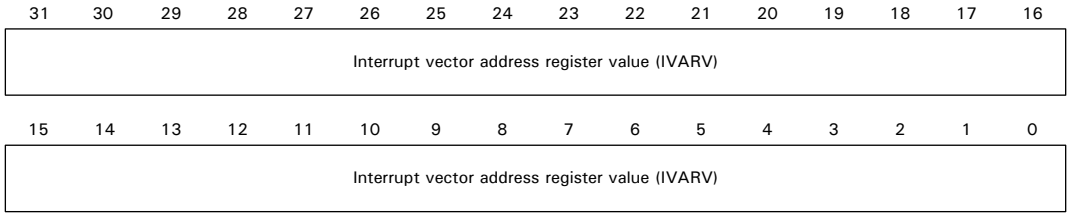
Downloaded from www.Manualslib.com manuals search engine

Interrupt Vector Address Register Level 31-0

Addresses: A090 00C4 (level 0) / 00C8 / 00CC / 00D0 / 00D4 / 00D8 / 00DC / 00E0 / 00E4 / 00E8 / 00EC / 00F0 / 00F4 / 00F8 / 00FC / 0100 / 0104 / 0108 / 010C / 0110 / 0114 / 0118 / 011C / 0120 / 0124 / 0128 / 012C / 0130 / 0134 / 0138 / 013C / 0140 (level 31)

The Interrupt Vector Address register configures the Interrupt vector address for each interrupt level source.

Register


31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16
Interrupt vector address register value (IVARV)
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Interrupt vector address register value (IVARV)

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:00	R/W	Int Vec Adr	0x0	**Interrupt Vector Address register** Interrupt vector address register bits.

Int (Interrupt) Config (Configuration) 31-0 registers

Addresses: A090 0144 / 0148 / 014C / 0150 / 0154 / 0158 / 015C / 0160

Each Interrupt Configuration register is 8 bits in length, and programs each interrupt configuration for each priority level.

Individual register mapping

This table shows how the 32 individual 8-byte registers are mapped in the eight 32-bit registers.

Register	[31:24]	[23:16]	[15:08]	[07:00]
A090 0144	Int Config 0	Int Config 1	Int Config 2	Int Config 3
A090 0148	Int Config 4	Int Config 5	Int Config 6	Int Config 7
A090 014C	Int Config 8	Int Config 9	Int Config 10	Int Config 11
A090 0150	Int Config 12	Int Config 13	Int Config14	Int Config 15
A090 0154	Int Config 16	Int Config 17	Int Config 18	Int Config 19
A090 0158	Int Config 20	Int Config 21	Int Config 22	Int Config 23

Downloaded from www.Manualslib.com manuals search engine

Register	[31:24]	[23:16]	[15:08]	[07:00]
A090 015C	Int Config 24	Int Config 25	Int Config 26	Int Config 27
A090 0160	Int Config 28	Int Config 29	Int Config 30	Int Config 31

Register bit assignment

This is how the bits are assigned in each register, using data bits [07:00] as the example.

Bits	Access	Mnemonic	Reset	Description
D07	R/W	IE	0x0	**Interrupt enable** 0 Interrupt is disabled 1 Interrupt is enabled
D06	R	INV	0x0	**Invert** 0 Do not invert the level of the interrupt source. 1 Invert the level of the interrupt source.
D05	R/W	IT	0x0	**Interrupt type** 0 IRQ 1 FIQ If FIQ is programmed, *Interrupt* must be the highest priority.
D04:00	R/W	ISD	0x0–0x1F	**Interrupt source ID** Assign an interrupt ID to each priority level. See "Interrupt sources," beginning on page 149, for the list of interrupt ID numbers.

ISADDR register

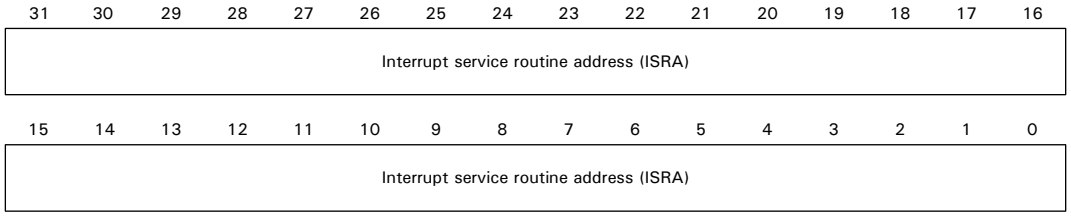
Address: A090 0164

The ISADDR register provides the current ISADDR value. Read and write to this register for IRQ interrupts only.

Immediately before the read to the ISADDR register, always perform an extra write or read to any other internal register to consume an extra clock cycle. Make sure that the extra access is not optimized away.

Downloaded from www.Manualslib.com manuals search engine

Register


31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16
Interrupt service routine address (ISRA)
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Interrupt service routine address (ISRA)

Register bit assignment

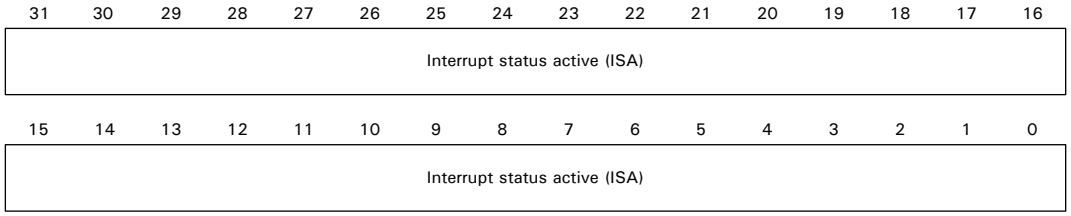
Bits	Access	Mnemonic	Reset	Description
D31:00	R/W	IS addr	0x0	**Interrupt service routine address** ■ A read to this register updates the priority logic block and masks the current and any lower priority interrupt requests. ■ Write the value of the interrupt level (0–31) to clear the current priority level.

Interrupt Status Active

Address: A090 0168

The Interrupt Status Active register shows the current active interrupt request.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:00	R	ISA	0x0	**Interrupt status active** Provides the status of all active, enabled interrupt request levels, where bit 0 is for the interrupt assigned to level 0, bit 1 is for the interrupt assigned to level 1, and so on through bit 31 for the interrupt assigned to level 31.

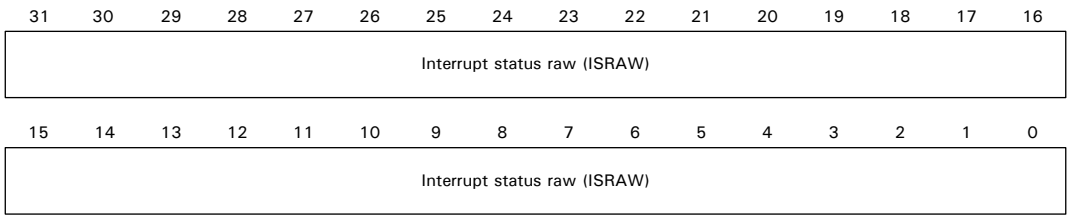
Downloaded from www.Manualslib.com manuals search engine

Interrupt Status Raw

Address: A090 016C

The Interrupt Status Raw register shows all current interrupt requests.

Register



Register bit assignment

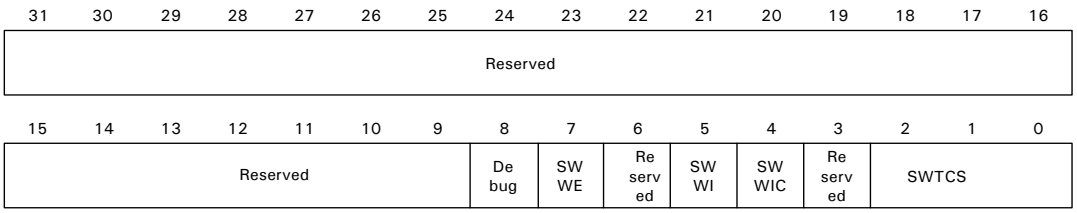
Bits	Access	Mnemonic	Reset	Description
D31:00	R	ISRAW	0x0	**Interrupt status raw** Provides the status of all active, enabled, and disabled interrupt request levels, where bit 0 is for the interrupt assigned to level 0, bit 1 is for the interrupt assigned to level 1, and so on through bit 31 for the interrupt assigned to level 31.

Software Watchdog Configuration

Address: A090 0174

The Software Watchdog Configuration register configures the software watchdog timer operation.

Register



Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:09	NA	Reserved	N/A	N/A
D08	R/W	Debug	0x0	**Debug mode** 0 Timer enabled in CPU debug mode 1 Timer disabled in CPU debug mode
D07	R/W	SWWE	0x0	**Software watchdog enable** 0 Software watchdog disabled 1 Software watchdog enabled; once set, cannot be cleared
D06	N/A	Reserved	N/A	N/A
D05	R/W	SWWI	0x0	**Software watchdog interrupt clear** Write a 1, then a 0 to this register to clear the software watchdog interrupt.
D04	R/W	SWWIC	0x0	**Software watchdog interrupt response** 0 Generate interrupt 1 Generate reset Note: If the interrupt option is selected and a software watchdog timeout occurs and the interrupt has not been cleared from a previous timeout, the reset is asserted.
D03	N/A	Reserved	N/A	N/A
D02:00	R/W	SWTCS	0x0	**Software watchdog timer clock select** 000 System memory clock / 2 001 System memory clock / 4 010 System memory clock / 8 011 System memory clock / 16 100 System memory clock / 32 101 System memory clock / 64 110 Reserved 111 Reserved

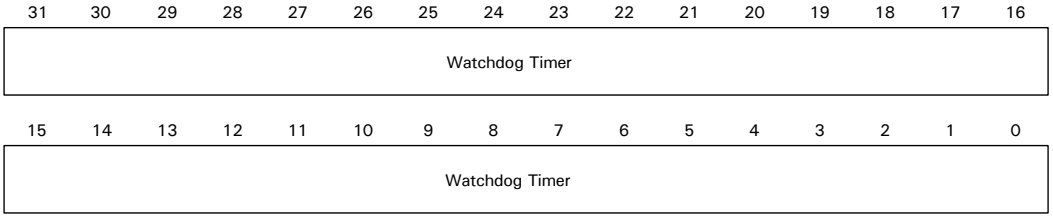
Software Watchdog Timer

Address: A090 0178

The Software Watchdog Timer register services the watchdog timer.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

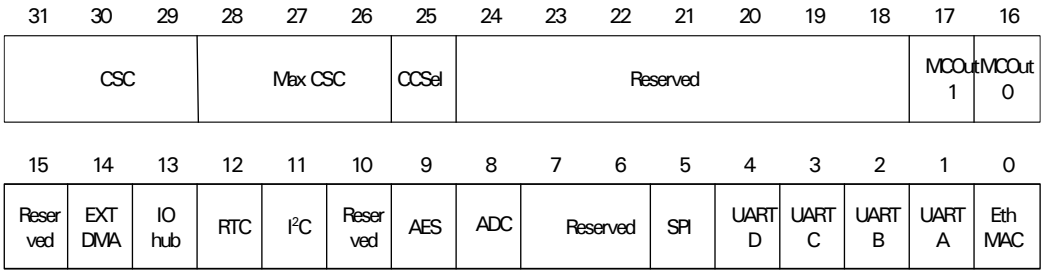
Bits	Access	Mnemonic	Reset	Description
D31:00	R/W	Watchdog timer	0x0	**Watchdog timer** ■ A read to this register gives the current value of the watchdog timer, but will not change the contents. ■ A write to the register changes the contents based on the write data value.

Clock Configuration register

Address: A090 017C

The Clock Configuration register enables and disables clocks to each module on the AHB bus.

Register



Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:29	R/W	CSC	0x000	**Clock scale control** 000 Full speed (149.9136/74.9568) 001 Divide by 2 (74.9568/37.4784) 010 Divide by 4 (37.4784/18.7392) 011 Divide by 8 (18.7393/9.3693) 100 Divide by 16 (9.3693/4.6848) Determines the frequency of the system clock rates. The full speed rate is 150MHz for the CPU clock and 75MHz for the AHB clock. If CCSEL = 0, then the CPU clock will be the same frequency as the AHB clock, 74.9568 maximum. This register can be written on the fly.
D28:26	R/W	Max CSC	0x000	**Max clock scale control** 000 Full speed (149.9136/74.9568) 001 Divide by 2 (74.9568/37.4784) 010 Divide by 4 (37.4784/18.7392) 011 Divide by 8 (18.7393/9.3693) 100 Divide by 16 (9.3693/4.6848) Software can write to the CSC bits to reduce the clock frequency of the CPU and AHB clocks. This register determines the maximum system CPU and AHB clock frequencies when returning low speed operation. This register is only valid if the hardware clock scale control bit is set in the Power Management register. If CCSEL = 0, then the CPU clock will be the same frequency as the AHB clock, 74.9568 maximum.
D25	R/W	CCSel	0x0	**CPU clock select** 0 CPU clock is equal to AHB clock 1 CPU clock is 2 x AHB clock
D24:18	N/A	Reserved	N/A	N/A
D17	R/W	MCOut 1	0x1	**Memory clock out 1** 0 Clock disabled 1 Clock enabled
D16	R/W	MCOut 0	0x1	**Memory clock out 0** 0 Clock disabled 1 Clock enabled
D15	N/A	Reserved	N/A	N/A
D14	R/W	EXT DMA	0x1	**External DMA** 0 Clock disabled 1 Clock enabled

Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Reset	Description
D13	R/W	IO hub	0x1	**IO hub** 0 Clock disabled 1 Clock enabled
D12	R/W	RTC	0x1	**RTC** 0 Clock disabled 1 Clock enabled
D11	R/W	I^2C	0x1	I^2C 0 Clock disabled 1 Clock enabled
D10	N/A	Reserved	N/A	N/A
D09	R/W	AES	0x0	**AES** 0 Clock disabled 1 Clock enabled
D08	R/W	ADC	0x1	**ADC** 0 Clock disabled 1 Clock enabled
D07:06	N/A	Reserved	N/A	Always write to 00
D05	R/W	SPI	0x1	**SPI** 0 Clock disabled 1 Clock enabled
D04	R/W	UART D	0x1	**UART D** 0 Clock disabled 1 Clock enabled
D03	R/W	UART C	0x1	**UART C** 0 Clock disabled 1 Clock enabled
D02	R/W	UART B	0x1	**UART B** 0 Clock disabled 1 Clock enabled
D01	R/W	UART A	0x1	**UART A** 0 Clock disabled 1 Clock enabled
D00	R/W	Eth MAC	0x1	**Ethernet MAC** 0 Clock disabled 1 Clock enabled

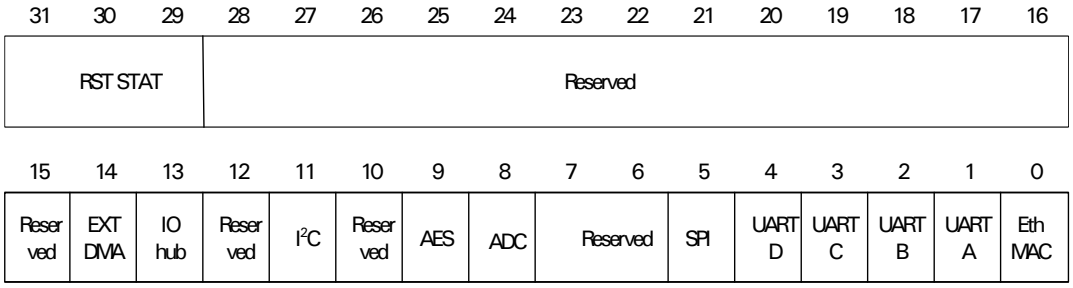
Module Reset register

Address: A090 0180

Downloaded from www.Manualslib.com manuals search engine

The Module Reset register resets each module on the AHB bus.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:29	R	RST STAT	Not reset	**Reset status** 001 External reset using reset_n 010 External reset using sreset_n 011 PLL change reset) 100 Software watchdog reset 101 AHB bus monitor reset Status to determine the cause of the last chip level reset.
D28:15	N/A	Reserved	N/A	N/A
D14	R/W	EXT DMA	0x1	**External DMA** 0 Module reset 1 Module enabled
D13	R/W	IO hub	0x1	**IO hub** 0 Module reset 1 Module enabled
D12	N/A	Reserved	N/A	N/A
D11	R/W	I^2C	0x1	I^2C 0 Module reset 1 Module enabled
D10	N/A	Reserved	N/A	N/A
D09	R/W	AES	0x0	**AES** 0 Module reset 1 Module enabled
D08	R/W	ADC	0x1	**ADC** 0 Module reset 1 Module enabled

Downloaded from www.Manualslib.com manuals search engine

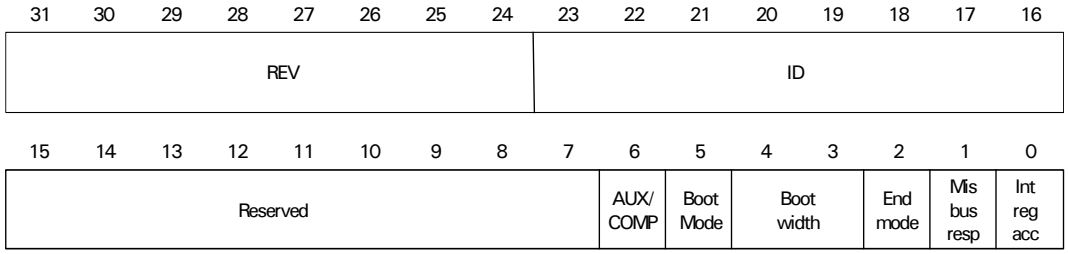
Bits	Access	Mnemonic	Reset	Description
D07:06	N/A	Reserved	N/A	Always write to 00
D05	R/W	SPI	0x1	**SPI** 0 Module reset 1 Module enabled
D04	R/W	UART D	0x1	**UART D** 0 Module reset 1 Module enabled
D03	R/W	UART C	0x1	**UART C** 0 Module reset 1 Module enabled
D02	R/W	UART B	0x1	**UART B** 0 Module reset 1 Module enabled
D01	R/W	UART A	0x1	**UART A** 0 Module reset 1 Module enabled
D00	R/W	Eth MAC	0x1	**Ethernet MAC** 0 Module reset 1 Module enabled

Miscellaneous System Configuration and Status register

Address: A090 0184

The Miscellaneous System Configuration and Status register configures miscellaneous system configuration bits.

Register


31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16
REV
ID
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Reserved
AUX/ COMP
Boot Mode
Boot width
End mode
Mis bus resp
Int reg acc

Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:24	R	REV	0x0	**Revision** Indicates the hardware identification and revision of the processor chip.
D23:16	R	ID	0x3	**Identification** Identifies the chip as: 0 NS9750B-A1 1 NS9360 2 NS9210 3 NS9215
D15:07	N/A	Reserved	N/A	N/A
D06	R	AUX/COMP	N/A	**Auxiliary analog comparator status** 0 Level is below 2.4V 1 Level is above 2.4V
D05	R	Boot mode	HW strap gpio_a[2]	**Boot mode** 0 Boot from SPI 1 Boot from flash
D04:03	R	Boot width	HW strap gpio_a[0] addr[23]	If boot mode is set to boot from flash: 00 8-bit 01 32-bit 10 32-bit 11 16-bit If boot mode is set to boot from SPI: 00 Reserved 01 Boot using 8-bit address SPI device 10 Boot using 24-bit address SPI device 11 Boot using 16-bit address SPI device
D02	R/W	End mode	HW strap gpio_a[3]	**Endian mode** 0 Little endian mode 1 Big endian mode
D01	R/W	Mis bus resp	0x0	**Misaligned bus address response mode** 0 Allow misaligned bus addresses 1 Generate an error response when a misaligned bus address is found; that is, when haddr bits 1 or 0 are not level 0.
D00	R/W	Int reg acc	0x1	**Internal register access mode bit 0** 0 Allow access to internal registers using PRIVILEGED mode only 1 Allow access to internal registers using PRIVILEGED or USER mode

Downloaded from www.Manualslib.com manuals search engine

PLL Configuration register

Address: A090 0188

The PLL Configuration register configures the PLL. A write to this register reconfigures and resets the PLL.

PLL frequency formula

This is the formula for PLL frequency:

PLL Vco = (RefClk / NR+1) * NF+1

ClkOut = PLL Vco / OD+1

Restrictions:

- (RefClk / NR+1) range: 275KHz–550MHz
- PLL Vco range: 110MHz–550MHz

Register

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:17	N/A	Reserved	N/A	N/A
D16:08	R/W	NF	0x3c	PLL feedback divider
D07	R/W	BP	HW strap ~addr[7]	**PLL bypass** 0 PLL enabled 1 PLL bypassed
D06:05	R/W	OD	HW strap ~addr [6:5]	PLL output divider
D04:00	R/W	NR	HW strap ~addr [4:3], addr[2:0]	PLL reference clock divider

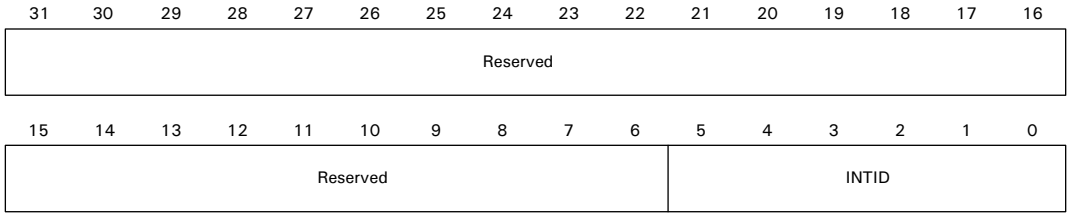
Downloaded from www.Manualslib.com manuals search engine

Active Interrupt Level ID Status register

Address: A090 018C

The Active Interrupt Level ID Status register is six bits in length, and shows the current active interrupt level ID.

Register



Register bit assignment

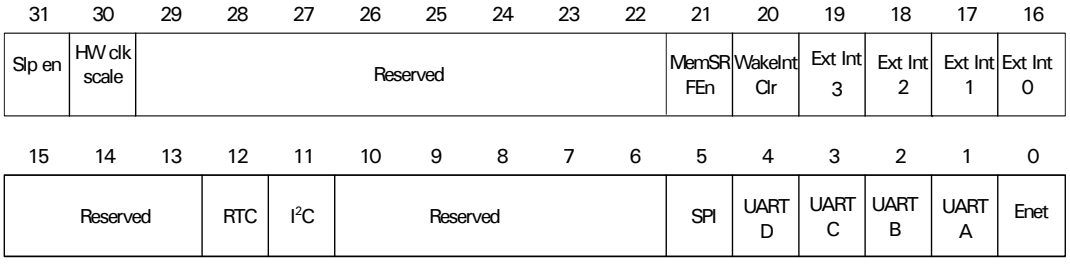
Bits	Access	Mnemonic	Reset	Description
D31:06	N/A	Reserved	N/A	N/A
D05:00	R	INTID	0x0	**Interrupt ID** The level ID of the current active interrupt.

Power Management

Address: A090 0228

The power management register controls the processor power management features.

Register



Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31	R/W	Slp en	0x0	**Deprecated Chip sleep enable** This control bit is provided for backwards compatibility with software written for the NS9750 and NS9360 processors, and should not be used by new software. System software writes a 1 to this bit to stop the clock to the CPU. Note that software is responsible for stopping the clocks to all other modules except the wakeup module(s) before setting this bit. When this bit is set, the clock to the CPU is stopped and the CPU is held in reset. **New designs should not use this bit.** They should stop the clock by executing the following coprocessor instruction: MCR p15, 0<Rd>, c7, c0, 4 This instruction places the ARM9 CPU into *wait for interrupt* mode. In *wait for interrupt* mode, the clock is stopped to the CPU but reset is not asserted. The CPU resumes and executes a CPU Wake Interrupt when activity is detected by one of the wakeup modules selected by the other bits in this register. The PC will be restored to the address after the coprocessor instruction that stopped the CPU's clock when the CPU Wake Interrupt ISR completes. The processor can not wake up on a timer interrupt because the system timers are stopped when the processor enters wake for interrupt mode.
D30	R/W	HW clk scale	0x0	**Hardware clock scale control** 0 Disable hardware clock scale control 1 Enable hardware clock scale control Used by hardware to increase the clock rate when activity is found on one of the modules enabled as a wakeup module. Hardware automatically increases the system clock frequencies to the value set by the max clock scale control bit in the Clock Control register.
D29:22	N/A	Reserved	N/A	N/A
D21	R/W	MemSRFEn	0x0	**SDRAM self refresh control** 0 Memory self refresh control disabled 1 Memory self refresh control enabled When enabled, the memory controller is automatically placed in self refresh mode when the CPU is in sleep mode and taken out of self refresh upon wakeup.

Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Reset	Description
D20	R/W	WakeIntClr	0x0	**CPU wake interrupt clear** Write a 1, followed by a 0 to clear the CPU wake interrupt.
D19	R/W	Ext Int 3	0x0	External interrupt 3 interrupt wakeup 0 Do not wake on external 3 interrupt 1 Wake on external 3 wakeup
D18	R/W	Ext Int 2	0x0	External interrupt 2 interrupt wakeup 0 Do not wake on external 2 interrupt 1 Wake on external 2 wakeup
D17	R/W	Ext Int 1	0x0	External interrupt 1 interrupt wakeup 0 Do not wake on external 1 interrupt 1 Wake on external 1 wakeup
D16	R/W	Ext Int 0	0x0	External interrupt 0 interrupt wakeup 0 Do not wake on external 0 interrupt 1 Wake on external 0 wakeup
D15:13	N/A	Reserved	N/A	N/A
D12	R/W	RTC	0x0	**RTC wakeup** 0 Do not wake on RTC interrupt 1 Wake on RTC interrupt
D11	R/W	I^2C	0x0	**I2C wakeup** 0 Do not wake on I^2C activity 1 Wake on I^2C activity
D10:06	N/A	Reserved	N/A	N/A
D05	R/W	SPI	0x0	**SPI wakeup** 0 Do not wake on SPI activity 1 Wake on SPI activity
D04	R/W	UART D	0x0	**UART D wakeup** 0 Do not wake on character match 1 Wake on character match
D03	R/W	UART C	0x0	**UART C wakeup** 0 Do not wake on character match 1 Wake on character match
D02	R/W	UART B	0x0	**UART B wakeup** 0 Do not wake on character match 1 Wake on character match

Downloaded from www.Manualslib.com manuals search engine

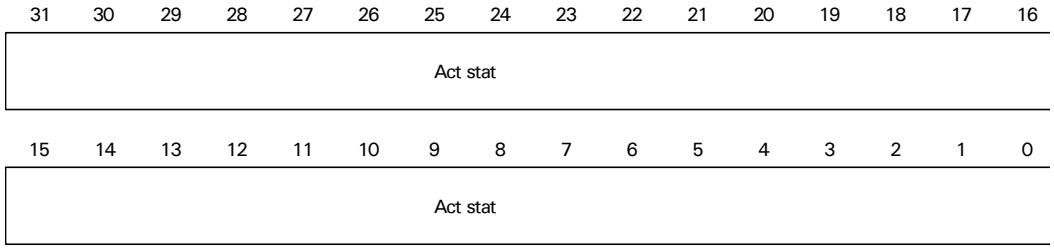
Bits	Access	Mnemonic	Reset	Description
D01	R/W	UART A	0x0	**UART A wakeup** 0 Do not wake on character match 1 Wake on character match
D00	R/W	Enet	0x0	**Ethernet wakeup** 0 Do not wake on Ethernet packet 1 Wake on Ethernet packet

AHB Bus Activity Status

Address: A090 022C

The AHB Bus Activity Status register is a read-only register that determines the activity on the AHB bus.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:00	R	Act stat	0x0	**Bus activity status** Provides the CPU with the status of activity on the system bus, excluding the CPU. This register can be used to help determine when to enter sleep mode or to reduce the system clock frequencies. The counter is reset each time a master accesses the AHB bus. The counter will saturate at all 1s.

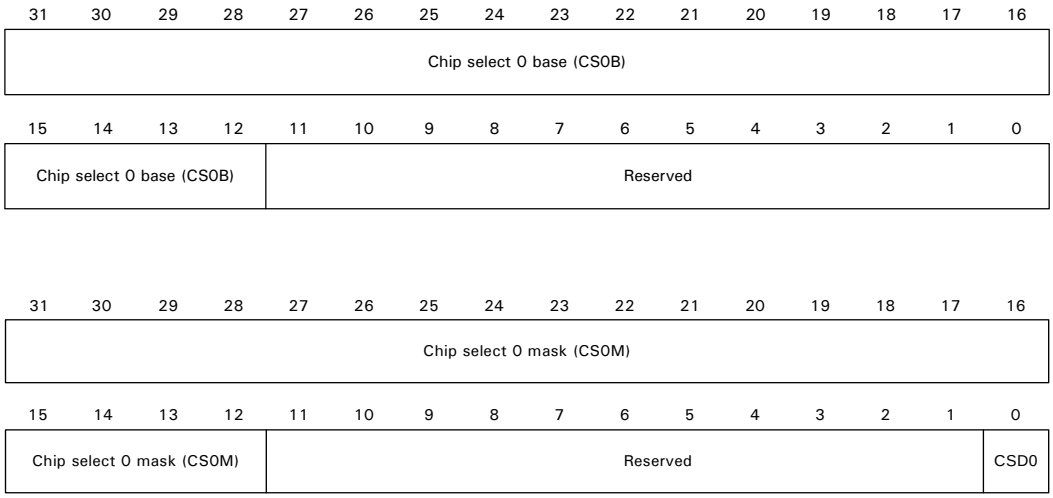
System Memory Chip Select 0 Dynamic Memory Base and Mask registers

Addresses: A090 01D0 / 01D4

Downloaded from www.Manualslib.com manuals search engine

These control registers set the base and mask for system memory chip select 0, with a minimum size of 4K. The powerup default settings produce a memory range of 0x0000 0000 — 0x0FFF FFFF.

Registers



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:12	R/W	CS0B	0x00000	**Chip select 0 base** Base address for chip select 0
D11:00	N/A	Reserved	N/A	N/A
D31:12	R/W	CS0M	0xF0000	**Chip select 0 mask** Mask or size for chip select 0
D11:01	N/A	Reserved	N/A	N/A
D00	R/W	CSD0	0x1	**Chip select 0 disable** 0 Disable chip select 1 Enable chip select

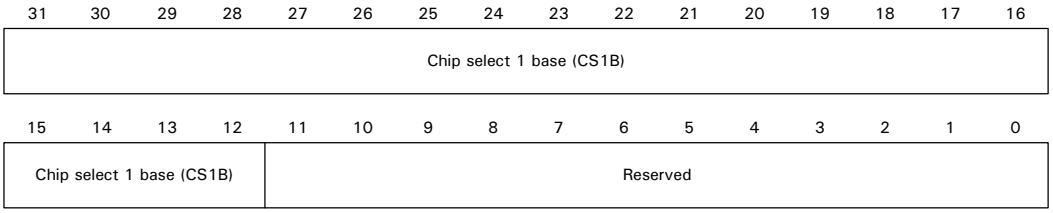
System Memory Chip Select 1 Dynamic Memory Base and Mask registers

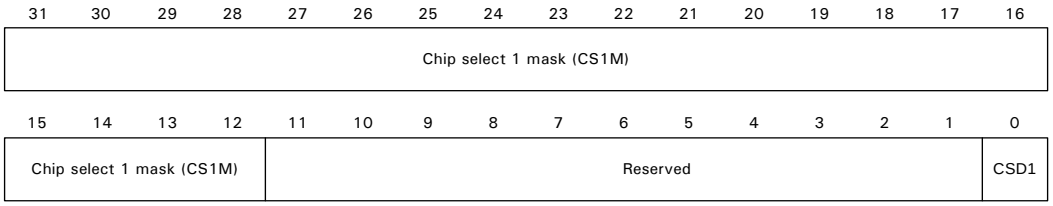
Addresses: A090 01D8 / 01DC

These control registers set the base and mask for system memory chip select 1, with a minimum size of 4K. The powerup default settings produce a memory range of 0x1000 0000 — 0x1FFF FFFF.

Downloaded from www.Manualslib.com manuals search engine

Registers





Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:12	R/W	CS1B	0x10000	**Chip select 1 base** Base address for chip select 1
D11:00	N/A	Reserved	N/A	N/A
D31:12	R/W	CS1M	0xF0000	**Chip select 1 mask** Mask or size for chip select 5
D11:01	N/A	Reserved	N/A	N/A
D00	R/W	CSD1	0x1	**Chip select 1disable** 0 Disable chip select 1 Enable chip select

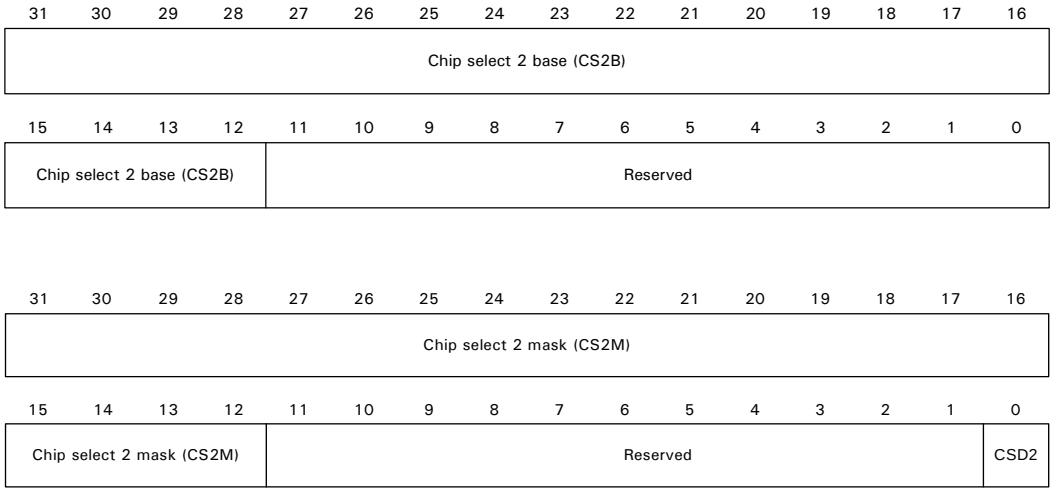
System Memory Chip Select 2 Dynamic Memory Base and Mask registers

Addresses: A090 01E0 / 01E4

These control registers set the base and mask for system memory chip select 2, with a minimum size of 4K. The powerup default settings produce a memory range of 0x2000 0000 — 0x2FFF FFFF.

Downloaded from www.Manualslib.com manuals search engine

Registers



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:12	R/W	CS2B	0x20000	**Chip select 2 base** Base address for chip select 2
D11:00	N/A	Reserved	N/A	N/A
D31:12	R/W	CS2M	0xF0000	**Chip select 2 mask** Mask or size for chip select 2
D11:01	N/A	Reserved	N/A	N/A
D00	R/W	CSD2	0x1	**Chip select 2 disable** 0 Disable chip select 1 Enable chip select

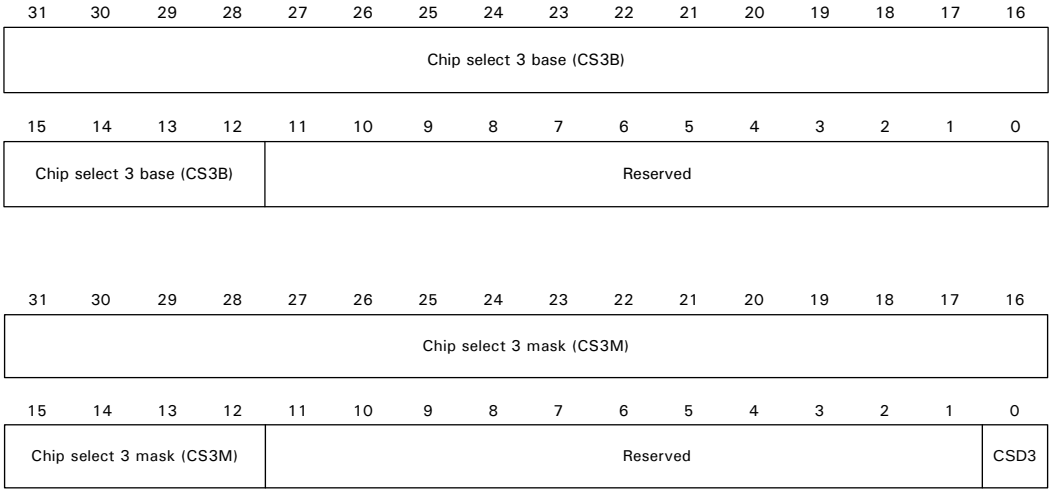
System Memory Chip Select 3 Dynamic Memory Base and Mask registers

Addresses: A090 01E8 / 01EC

These control registers set the base and mask for system memory chip select 3, with a minimum size of 4K. The powerup default settings produce a memory range of 0x3000 0000 — 0x3FFF FFFF.

Downloaded from www.Manualslib.com manuals search engine

Registers



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:12	R/W	CS3B	0x30000	**Chip select 3 base** Base address for chip select 3
D11:00	N/A	Reserved	N/A	N/A
D31:12	R/W	CS3M	0xF0000	**Chip select 3 mask** Mask or size for chip select 3
D11:01	N/A	Reserved	N/A	N/A
D00	R/W	CSD3	0x1	**Chip select 3 disable** 0 Disable chip select 1 Enable chip select

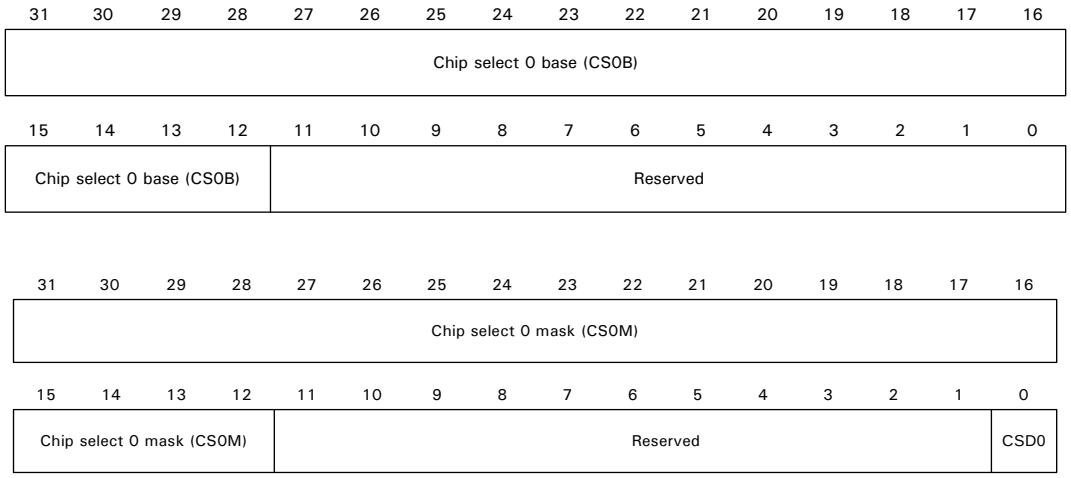
System Memory Chip Select 0 Static Memory Base and Mask registers

Addresses: A090 01F0 / 01F4

These control registers set the base and mask for system memory chip select 0, with a minimum size of 4K. The powerup default settings produce a memory range of 0x4000 0000 — 0x4FFF FFFF.

Downloaded from www.Manualslib.com manuals search engine

Registers



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:12	R/W	CS0B	0x40000	**Chip select 0 base** Base address for chip select 0.
D11:00	N/A	Reserved	N/A	N/A
D31:12	R/W	CS0M	0xF0000	**Chip select 0 mask** Mask or size for chip select 0.
D11:01	N/A	Reserved	N/A	N/A
D00	R/W	CSD0	0x1	**Chip select 0 disable** 0 Disable chip select 1 Enable chip select

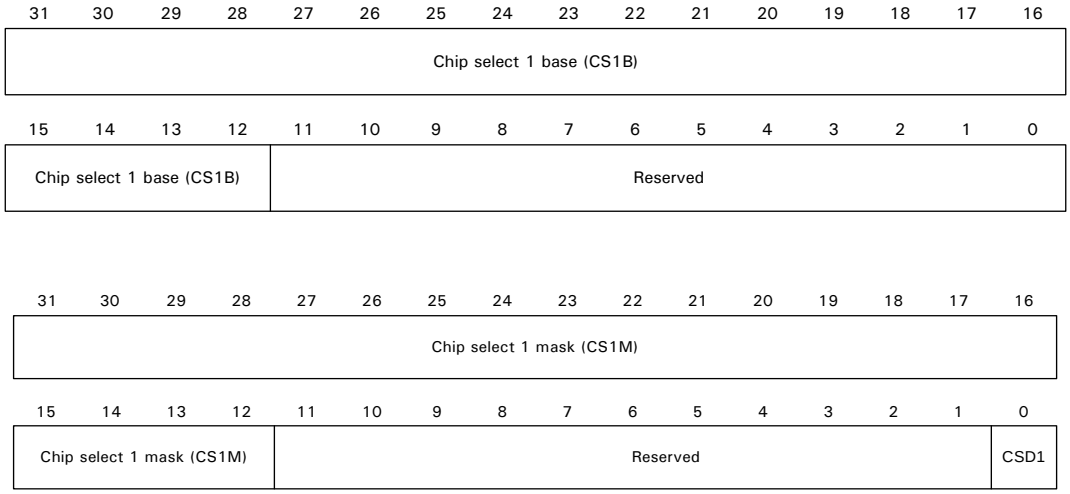
System Memory Chip Select 1 Static Memory Base and Mask registers

Addresses: A09001F8 / 01FC

These control registers set the base and mask for system memory chip select 1, with a minimum size of 4K. The powerup default settings produce a memory range of 0x5000 0000 — 0x5FFF FFFF.

Downloaded from www.Manualslib.com manuals search engine

Registers



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:12	R/W	CS1B	0x50000	**Chip select 1 base** Base address for chip select 1
D11:00	N/A	Reserved	N/A	N/A
D31:12	R/W	CS1M	0xF0000	**Chip select 1 mask** Mask or size for the chip select 1.
D11:01	N/A	Reserved	N/A	N/A
D00	R/W	CSD1	0x1	**Chip select 1 disable** 0 Disable chip select 1 Enable chip select

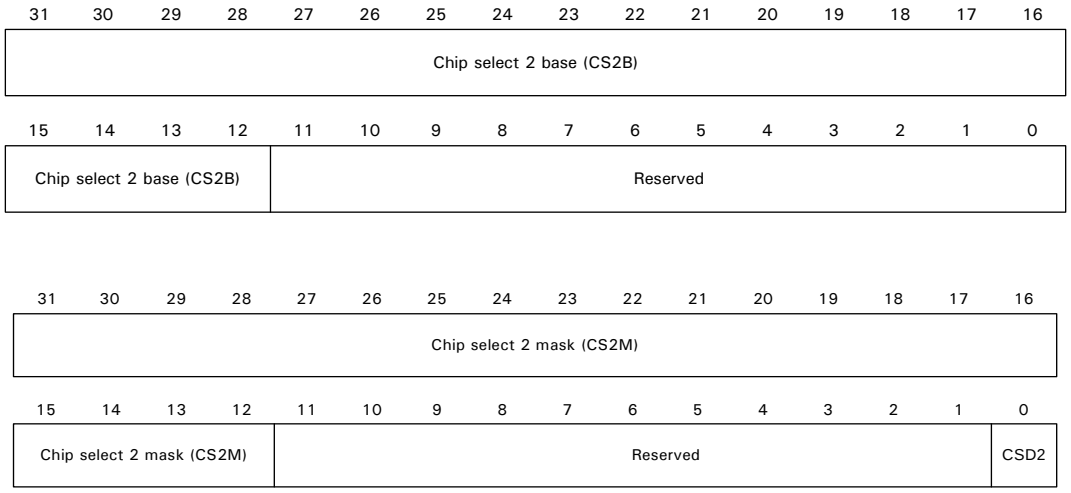
System Memory Chip Select 2 Static Memory Base and Mask registers

Addresses: A090 0200 / 0204

These control registers set the base and mask for system memory chip select 2, with a minimum size of 4K. The powerup default settings produce a memory range of 0x6000 0000 — 0x6FFF FFFF.

Downloaded from www.Manualslib.com manuals search engine

Registers



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:12	R/W	CS2B	0x60000	**Chip select 2 base** Base address for chip select 2.
D11:00	N/A	Reserved	N/A	N/A
D31:12	R/W	CS2M	0xF0000	**Chip select 2 mask** Mask or size for chip select 2.
D11:01	N/A	Reserved	N/A	N/A
D00	R/W	CSD2	0x1	**Chip select 2 disable** 0 Disable chip select 1 Enable chip select

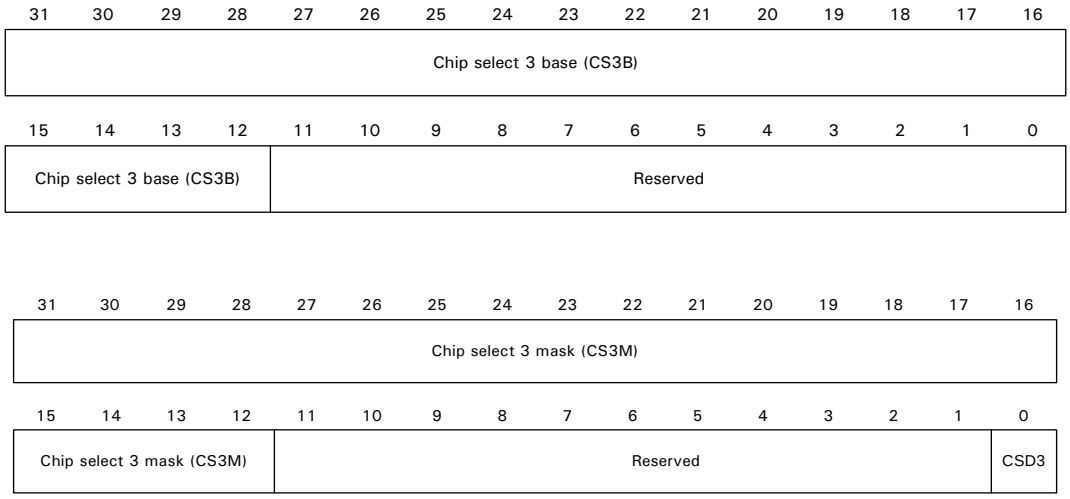
System Memory Chip Select 3 Static Memory Base and Mask registers

Addresses: A090 0208 / 020C

These control registers set the base and mask for system memory chip select 3, with a minimum size of 4K. The powerup default settings produce a memory range of 0x7000 0000 — 0x7FFF FFFF.

Downloaded from www.Manualslib.com manuals search engine

Registers



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:12	R/W	CS3B	0x70000	**Chip select 3 base** Base address for chip select 3.
D11:00	N/A	Reserved	N/A	N/A
D31:12	R/W	CS3M	0xF0000	**Chip select 3 mask** Mask or size for chip select 3.
D11:01	N/A	Reserved	N/A	N/A
D00	R/W	CSD3	0x1	**Chip select 3 disable** 0 Disable chip select 1 Enable chip select

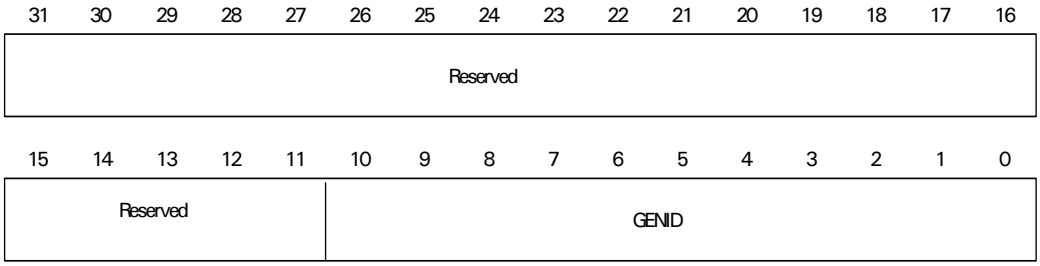
Gen ID register

Address: A090 0210

This register is read-only, and indicates the state of addr[19:09] pins at powerup.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

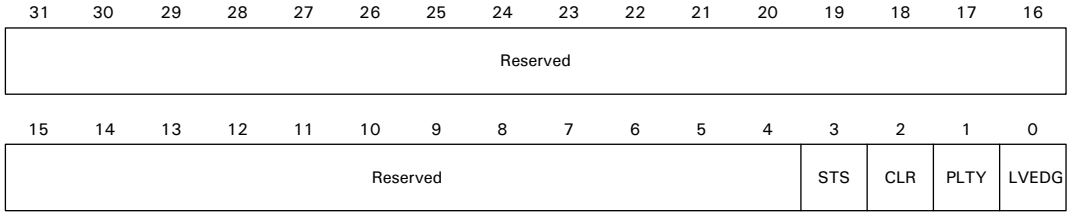
Bits	Access	Mnemonic	Reset	Description
D31:11	N/A	Reserved	N/A	**N/A**
D10:00	R	GENID	HW strap addr[19:09]	General Purpose ID register

External Interrupt 0-3 Control register

Addresses: A090 0214 / 0218 / 021C / 0220

The External Interrupt Control registers control the behavior of external interrupts 0–3.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:04	N/A	Reserved	N/A	N/A
D03	R	STS	N/A	**Status** Status of the external signal before edge detect or level conversion.
D02	R/W	CLR	0x0	**Clear** Write a 1, then a 0 to this bit to clear the interrupt generated by the edge detect circuit.

Downloaded from www.Manualslib.com manuals search engine

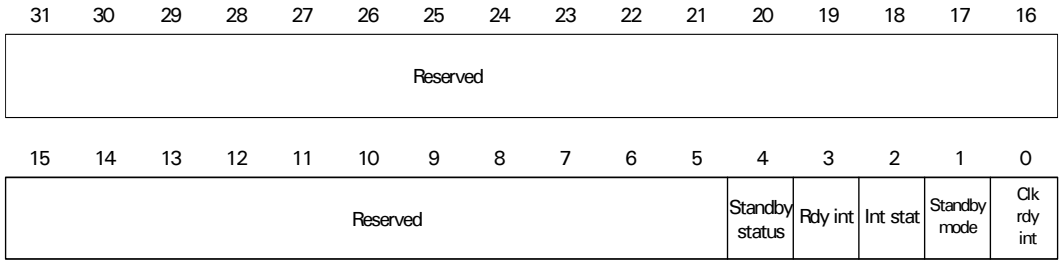
Bits	Access	Mnemonic	Reset	Description
D01	R/W	PLTY	0x0	**Polarity** 0 If level-sensitive, the input source is active high. If edge-sensitive, generate an interrupt on the rising edge of the external interrupt. 1 If level-sensitive, the input source is active low. The level is inverted before sending to the interrupt controller. If edge-sensitive, generate an interrupt on the falling edge of the external interrupt.
D00	R/W	LVEDG	0x0	**Level edge** 0 Level-sensitive interrupt 1 Edge-sensitive interrupt

RTC Module Control register

Address: A090 0224

The RTC Module Control register controls the RTC module.

Register


31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16
Reserved
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Reserved
Standby status
Rdy int
Int stat
Standby mode
Clk rdy int

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:05	N/A	Reserved	N/A	N/A
D04	R	Standby status	0x0	**RTC standby mode status** 0 RTC module is in standby mode and cannot be accessed by the CPU. 1 RTC module is not in standby mode and can be accessed by the CPU.

Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Reset	Description
D03	R	Rdy int	0x0	**RTC clock ready interrupt status** 0 RTC clock ready interrupt not asserted 1 RTC clock ready interrupt asserted Note: The RTC clock ready and RTC module interrupts are ORed together to the interrupt controller. Read this bit to determine the actual source.
D02	R	Int stat	0x0	**RTC module interrupt status** 0 RTC module interrupt not asserted 1 RTC module interrupt asserted Note: The RTC clock ready and RTC module interrupts are ORed together to the interrupt controller. Read this bit to determine the actual source.
D01	R/W	Standby mode	0x0	**RTC standby mode** Allows the RTC module to be placed in low power mode. 0 The RTC module is placed in standby mode and cannot be accessed by the CPU. The RTC clock must be enabled when in standby mode (bit 10). 1 Normal operation. The CPU must wait for the RTC interrupt and read the status to determine that the clock change is complete (RTC clock ready interrupt status bit is set). The clock change may take up to 30 microseconds after this bit is set. Note: This bit must be set to 0 when not accessing the RTC registers or battery back RAM. When early power loss interrupt is detected, set this bit to 0.
D00	R/W	Clk rdy int	0x0	**RTC clock ready interrupt clear** 0 RTC clock ready interrupt enabled 1 RTC clock ready interrupt cleared Note: This register must be set, then cleared to service the RTC clock ready interrupt.

Downloaded from www.Manualslib.com manuals search engine

Downloaded from www.Manualslib.com manuals search engine

Memory Controller

CHAPTER 5

The Multiport Memory Controller is an AMBA-compliant system-on-chip (SoC) peripheral that connects to the Advanced High-performance Bus (AHB). The remainder of this chapter refers to this controller as the *memory controller*.

Features

The memory controller provides these features:

- AMBA 32-bit AHB compliancy
- Dynamic memory interface support including SDRAM and JEDEC low-power SDRAM
- Asynchronous static memory device support including RAM, ROM, and Flash, with and without asynchronous page mode
- Can operate with cached processors with copyback caches
- Can operate with uncached processors
- Low transaction latency
- Read and write buffers to reduce latency and improve performance, particularly for uncached processors.
- 8-bit, 16-bit, and 32-bit wide static memory support.
- 16-bit and 32-bit wide chip select SDRAM memory support.
- Static memory features, such as:
  - Asynchronous page mode read
  - Programmable wait states
  - Bus turnaround delay
  - Output enable and write enable delays
  - Extended wait

Downloaded from www.Manualslib.com manuals search engine

- Power-saving modes that dynamically control SDRAM clk_en.
- Dynamic memory self-refresh mode supported by a power management unit (PMU) interface or by software.
- Controller supports 2K, 4K, and 8K row address synchronous memory parts; that is, typical 512 MB, 256 MB, and 16 Mb parts with 8, 16, or 32 DQ bits per device.
- A separate AHB interface to program the memory controller. This enables the memory controller registers to be situated in memory with other system peripheral registers.
- Locked AHB transaction support.
- Support for all AHB burst types.
- Little and big endian support.

Note: Synchronous static memory devices (synchronous burst mode) are not supported.

Low-power operation

In many systems, the contents of the memory system have to be maintained during low-power sleep modes. The processor provides two features to enable this:

- Dynamic memory refresh over soft reset
- A mechanism to place the dynamic memories into self-refresh mode

Self-refresh mode can be entered as follows:

1 Set the SREFREQ bit in the Dynamic Memory Control register.

2 Poll the SREFACK bit in the Status register.

Note: Static memory can be accessed as normal when the SDRAM memory is in self-refresh mode.

Low-power SDRAM deep-sleep mode

The memory controller supports JEDEC low-power SDRAM deep-sleep mode. Deep-sleep mode can be entered by setting the deep-sleep (DP) bit in the Dynamic Memory Control register. The device is put into a low-power mode where it is powered down and no longer refreshed. All data in the memory is lost.

Low-power SDRAM partial array refresh

The memory controller supports JEDEC low-power SDRAM partial array refresh. Partial array refresh can be programmed by initializing the SDRAM memory device appropriately. When the memory device is put into self-refresh mode, only the memory banks specified are refreshed. The memory banks that are not refreshed lose their data contents.

Downloaded from www.Manualslib.com manuals search engine

Memory map

The memory controller provides hardware support for booting from external nonvolatile memory. During booting, the nonvolatile memory must be located at address 0x00000000 in memory. When the system is booted, the SRAM or SDRAM memory can be remapped to address 0x00000000 by modifying the address map in the AHB decoder.

Power-on reset memory map

On power-on reset, memory chip select 1 is mirrored onto memory chip select 0 and chip select 4. Any transactions to memory chip select 0 or chip select 4 (or chip select 1), then, access memory chip select 1. Clearing the address mirror bit (M) in the Control register disables address mirroring, and memory chip select 0, chip select 4, and memory chip select 1 can be accessed as normal.

Chip select 1 memory configuration

You can configure the memory width and chip select polarity of static memory chip select 1 by using selected input signals. This allows you to boot from chip select 1.

These are the bootstrap signals:

- gpio_a[0], addr[23]: Memory width select
- gpio_a[2]: Boot mode

Example: Boot from flash, SRAM mapped after boot

The system is set up as:

- Chip select 1 is connected to the boot flash device.
- Chip select 0 is connected to the SRAM to be remapped to 0x00000000 after boot.

This is the boot sequence:

1 At power-on, the reset chip select 1 is mirrored into chip select 0 (and chip select 4).

Downloaded from www.Manualslib.com manuals search engine

2 When the power-on reset (reset_n) goes inactive, the processor starts booting from 0x00000000 in memory.

3 The software programs the optimum delay values in the flash memory so the boot code can run at full speed.

4 The code branches to chip select 1 so the code can continue executing from the non-remapped memory location.

5 The appropriate values are programmed into the memory controller to configure chip select 0.

6 The address mirroring is disabled by clearing the address mirror (M) field in the Control register.

7 The ARM reset and interrupt vectors are copied from flash memory to SRAM that can then be accessed at address 0x00000000.

8 More boot, initialization, or application code is executed.

Example: Boot from flash, SDRAM remapped after boot

The system is set up as:

- Chip select 1 is connected to the boot flash device.
- Chip select 4 is connected to the SDRAM to be remapped to 0x00000000 after boot.

This is the boot sequence:

1 At power-on, the reset chip select 1 is mirrored into chip select 4 (and chip select 0).

Downloaded from www.Manualslib.com manuals search engine

2 When the power-on reset (reset_n) goes inactive, the processor starts booting from 0x00000000 in memory.

3 The software programs the optimum delay values in flash memory so the boot code can run at full speed.

4 The code branches to chip select 1 so the code can continue executing from the non-remapped memory location.

5 The appropriate values are programmed into the memory controller to configure chip select 4, and the memory device is initialized.

6 The address mirroring is disabled by clearing the address mirror (M) field in the Control register.

7 The ARM reset and interrupt vectors are copied from flash memory to SDRAM that can then be accessed at address 0x00000000.

8 More boot, initialization, or application code is executed.

Static memory controller

This table shows configurations for the static memory controller with different types of memory devices. See “StaticMemory Configuration 0–3 registers” on page 251 for more information.

Device	Write protect	Page mode	Buffer
ROM	Enabled	Disabled	Disabled [a]
Page mode ROM	Enabled	Enabled	Enabled [a]
Extended wait ROM	Enabled	Disabled	Disabled [a]
SRAM	Disabled (or enabled) [b]	Disabled	Disabled [a]
Page mode SRAM	Disabled (or enabled) [b]	Enabled	Enabled [a]
Extended wait SRAM	Disabled (or enabled) [b]	Disabled	Disabled [a]
Flash	Disabled or (enabled) [b]	Disabled	Disabled [c]
Page mode flash	Disabled or (enabled) [b]	Enabled	Enabled [c]
Extended wait flash	Disabled or (enabled) [b]	Disabled	Disabled [a]
Memory mapped peripheral	Disabled (or enabled) [b]	Disabled	Disabled

a Enabling the buffers means that any access causes the buffer to be used. Depending on the application, this can provide performance improvements. Devices without async-page-mode support generally work better with the buffer disabled. Again, depending on the application, this can provide performance improvements.

b SRAM and Flash memory devices can be write-protected if required.

c Buffering must be disabled when performing Flash memory commands and during writes.

Downloaded from www.Manualslib.com manuals search engine

Notes:

- Buffering enables the transaction order to be rearranged to improve memory performance. If the transaction order is important, the buffers must be disabled.
- Extended wait and page mode cannot be enabled at the same time.

Write protection

Each static memory chip select can be configured for write-protection. SRAM usually is unprotected and ROM devices must be write-protected (to avoid potential bus conflict when performing a write access to ROM), but the P field in the Static Memory Configuration register (see "StaticMemory Configuration 0–3 registers" on page 251) can be set to write-protect SRAM as well as ROM devices. If a write access is made to a write-protected memory bank, a bus error occurs. If a write access is made to a memory bank containing ROM devices and the chip select is not write-protected. An error is not returned and the write access proceeds as normal. Note that this might lead to a bus conflict.

Extended wait transfers

The static memory controller supports extremely long transfer times. In normal use, the memory transfers are timed using the Static Memory Read Delay register (StaticWaitRd) and Static Memory Wait Delay register (StaticWaitWr). These registers allow transfers with up to 32 wait states. If a very slow static memory device has to be accessed, however, you can enable the static configuration extended wait (EW) bit. When EW is enabled, the Static Extended Wait register is used to time both the read and write transfers. The Static Extended Wait register allows transfers to have up to 16368 wait states.

A peripheral can, at any time, signal to the processor that it wants to complete an access early by asserting the ns_ta_strb signal. This allows a slow peripheral with variable access times to signal that it is ready to complete an access. The processor normally completes an access when it finds a rising edge on ns_ta_strb.

For a burst access, the peripheral must toggle ns_ta_strb for each access it wants to complete early. The peripheral is not required to assert ns_ta_strb for each access in the burst; for example, the peripheral requires the programmed access for the start of a four access burst followed by three early completion accesses, each signalled by the assertion of ns_ta_strb.

Using the ns_ta_strb signal is valid only when the EW bit is enabled.

Be aware:

- Using extremely long transfer times might mean that SDRAM devices are not refreshed correctly.
- Very slow transfers can degrade system performance, as the external memory interface is tied up for long periods of time. This has detrimental effects on

Downloaded from www.Manualslib.com manuals search engine

time critical services, such as interrupt latency and low latency devices; for example, video controllers.

Memory mapped peripherals

Some systems use external peripherals that can be accessed using the static memory interface. Because of the way many of these peripherals function, the read and write transfers to them must not be buffered. The buffer must therefore be disabled.

Static memory initialization

Static memory must be initialized as required after poweron reset (reset_n) by programming the relevant registers in the memory controller as well as the configuration registers in the external static memory device.

Access sequencing and memory width

The data width of each external memory bank must be configured by programming the appropriate bank configuration register (Static Memory Configuration 0–3). When the external memory bus is narrower that the transfer initiated from the current main bus master, the internal bus transfer takes several external bus transfers to complete.

For example, if bank 0 is configured as 8-bit wide memory and a 32-bit read is initiated, the AHB bus stalls while the memory controller reads four consecutive bytes from the memory. During these accesses, the static memory controller block demultiplexes the four bytes into one 32-bit word on the AHB bus.

Wait state generation

Each bank of the memory controller must be configured for external transfer wait states in read and write accesses.

Configure the banks by programming the appropriate bank control registers:

- “StaticMemory Configuration 0–3 registers” on page 251 (StaticConfig[n])
- “StaticMemory Write Enable Delay 0–3 registers” on page 254 (StaticWaitWen[n])
- “Static Memory Output Enable Delay 0–3 registers” on page 255 (StaticWaitOen[n])
- “Static Memory Read Delay 0–3 registers” on page 256 (StaticWaitRd[n])
- “Static Memory Write Delay 0–3 registers” on page 257 (StaticWaitWr[n])
- “StaticMemory Page Mode Read Delay 0–3 registers” on page 256 (StaticWaitPage[n])
- “StaticMemory Turn Round Delay 0–3 registers” on page 258 (StaticWaitTurn[n])

Downloaded from www.Manualslib.com manuals search engine

- "Static Memory Extended Wait register" on page 247 (StaticExtendedWait)

The number of cycles in which an AMBA transfer completes is controlled by two additional factors:

- Access width
- External memory width

Programmable enable

Each bank of the memory controller has a programmable enable for the extended wait (EW). The WAITRD wait state field in the Static Memory Read Delay register can be programmed to select from 1–32 wait states for read memory accesses to SRAM and ROM, or the initial read access to page mode devices. The WAITWR wait state field in the Static Memory Write Delay register can be programmed to select from 1–32 wait states for access to SRAM. The Static Memory Page Mode Read Delay register can be programmed to select from 1–32 wait states for page mode accesses.

Static memory read control

There are three types of static memory read controls:

- Output enable programmable delay
- ROM, SRAM, and flash
- Asynchronous page mode read

Output enable programmable delay

The delay between the assertion of the chip select and the output enable is programmable from 0 to 15 cycles using the wait output enable bits (WAITOEN) in the Static Memory Output Enable Delay registers. The delay is used to reduce power consumption for memories that cannot provide valid output data immediately after the chip select has been asserted. The output enable is always deasserted at the same time as the chip select, at the end of the transfer.

ROM, SRAM, and Flash

The memory controller uses the same read timing control for ROM, SRAM, and flash devices. Each read starts with the assertion of the appropriate memory bank chip select signals (cs_n) and memory address (addr[27:0]). The read access time is determined by the number of wait states programmed for the WAITRD field in the Static Memory Read Delay register. The WAITTURN field in the Static Memory Turn Round Delay register determines the number of bus turnaround wait states added between external read and write transfers.

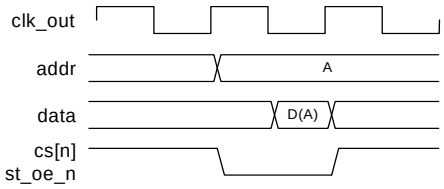
Downloaded from www.Manualslib.com manuals search engine

Static memory read: Timing and parameters

This section shows static memory read timing diagrams and parameters.

External memory read transfer with zero wait states

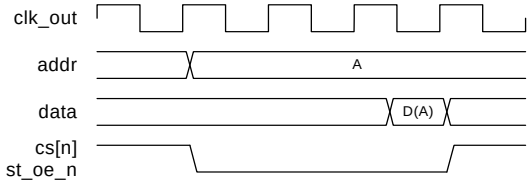
This diagram shows an external memory read transfer with the minimum zero wait states (WAITRD=0). Maximum performance is achieved when accessing the external device with load multiple (LDM) or store multiple (STM) CPU instructions.



Timing parameter	Value
WAITRD	0
WAITOEN	0
WAITPAGE	N/A
WAITWR	N/A
WAITWEN	N/A
WAITTURN	N/A

External memory read transfer with two wait states

This diagram shows an external memory read transfer with two wait states (WAITRD=2). Seven AHB cycles are required for the transfer, five for the standard read access and an additional two because of the programmed wait states added (WAITRD).



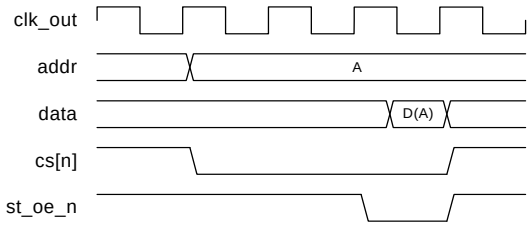
Timing parameter	Value
WAITRD	2
WAITOEN	0
WAITPAGE	N/A
WAITWR	N/A

Downloaded from www.Manualslib.com manuals search engine

Timing parameter	Value
WAITEN	N/A
WAITTURN	N/A

External memory read transfer with two output enable delay states

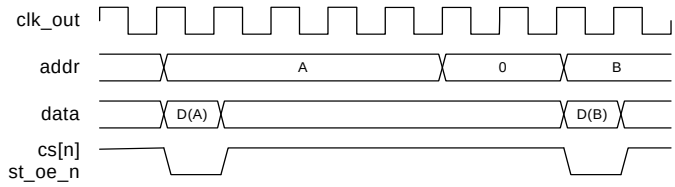
This diagram shows an external memory read transfer with two output enable delay states (WAITOEN=2). Seven AHB cycles are required for the transfer, five for the standard read and an additional two because of the output delay states added.



Timing parameter	Value
WAITRD	2
WAITOEN	2
WAITPAGE	N/A
WAITWR	N/A
WAITWEN	N/A
WAITTURN	N/A

External memory read transfers with zero wait states

This diagram shows external memory read transfers with zero wait states (WAITRD=0). These transfers can be non-sequential transfers or sequential transfers of a specified burst length. Bursts of unspecified length are interpreted as INCR4 transfers. All transfers are treated as separate reads, so have the minimum of five AHB cycles added.

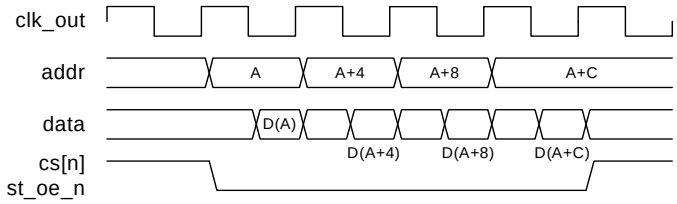


Downloaded from www.Manualslib.com manuals search engine

Timing parameter	Value
WAITRD	0
WAITOEN	0
WAITPAGE	N/A
WAITWR	N/A
WAITWEN	N/A
WAITTURN	N/A

Burst of zero wait states with fixed length

This diagram shows a burst of zero wait state reads with the length specified. Because the length of the burst is known, the chip select can be held asserted during the whole burst and generate the external transfers before the current AHB transfer has completed. The first read requires five arbitration cycles; the three subsequent sequential reads have zero AHB cycles added because the external transfers are automatically generated.

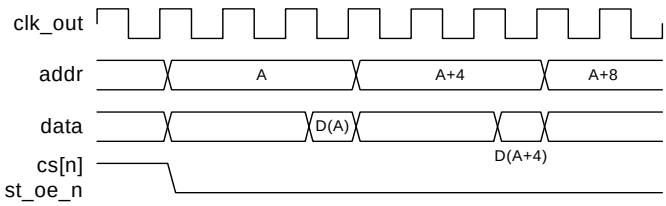


Timing parameter	Value
WAITRD	0
WAITOEN	0
WAITPAGE	N/A
WAITWR	N/A
WAITWEN	N/A
WAITTURN	N/A

Burst of two wait states with fixed length

This diagram shows a burst of two wait state reads with the length specified. The WAITRD value is used for all transfers in the burst.

Downloaded from www.Manualslib.com manuals search engine



Timing parameter	Value
WAITRD	2
WAITOEN	0
WAITPAGE	N/A
WAITWR	N/A
WAITWEN	N/A
WAITTURN	N/A

Asynchronous page mode read

The memory controller supports asynchronous page mode read of up to four memory transfers by updating address bits addr[1] and addr[0]. This feature increases the bandwidth by using a reduced access time for the read accesses that are in page mode. The first read access takes static wait read and WAITRD cycles. Subsequent read accesses that are in page mode take static wait page and WAITPAGE cycles. The chip select and output enable lines are held during the burst, and only the lower two address bits change between subsequent accesses. At the end of the burst, the chip select and output enable lines are deasserted together.

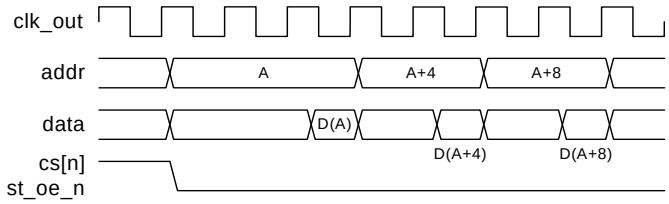
Asynchronous page mode read: Timing and parameters

This section shows asynchronous page mode read timing diagrams and parameters.

External memory page mode read transfer

Thls diagram shows an external memory page mode read transfer with two initial wait states and one sequential wait state. The first read requires five AHB arbitration cycles (plus three wait states); the following (up to 3) sequential transfers have only one AHB wait state. This gives increased performance over the equivalent nonpage mode ROM timing.

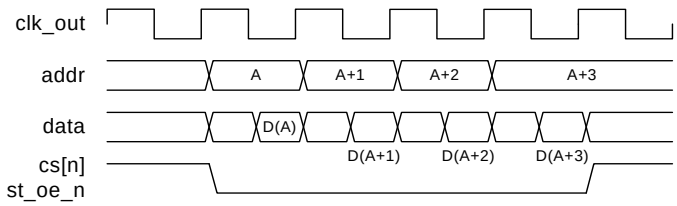
Downloaded from www.Manualslib.com manuals search engine



Timing parameter	Value
WAITRD	2
WAITOEN	0
WAITPAGE	1
WAITWR	N/A
WAITWEN	N/A
WAITTURN	N/A

External memory 32-bit burst read from 8-bit memory

This diagram shows a 32-bit read from an 8-bit page mode ROM device, causing four burst reads to be performed. A total of eight AHB wait states are added during this transfer, five AHB arbitration cycles and then one for each of the subsequent reads. WAITRD and WAITPAGE are 0.



Timing parameters	Value
WAITRD	0
WAITOEN	0
WAITPAGE	0
WAITWR	N/A
WAITWEN	N/A
WAITTURN	N/A

Downloaded from www.Manualslib.com manuals search engine

Static memory write control

Write enable programming delay

The delay between the assertion of the chip select and the write enable is programmable from 1 to 16 cycles using the WAITWEN bits of the Static Memory Write Enable Delay (StaticWaitWen[3:0]) registers. The delay reduces the power consumption for memories. The write enable is asserted on the rising edge of HCLK after the assertion of the chip select for zero wait states. The write enable is always deasserted a cycle before the chip select, at the end of the transfer. datamask_n (byte lane signal) has the same timing as st_we_n (write enable signal) for writes to 8-bit devices that use the byte lane selects instead of the write enables.

SRAM

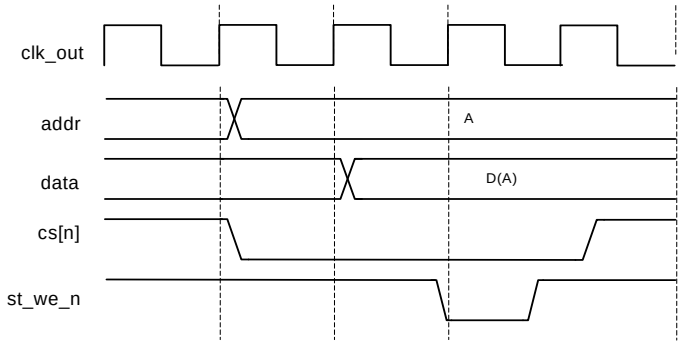
Write timing for SRAM starts with assertion of the appropriate memory bank chip selects (cs[n]_n) and address signals (addr[27:0]_n). The write access time is determined by the number of wait states programmed for the WAITWR field in the Static Memory Write Delay register (see “Static Memory Write Delay 0–3 registers” on page 257). The WAITTURN field in the bank control register (see “StaticMemory Turn Round Delay 0–3 registers” on page 258) determines the number of bus turnaround wait states added between external read and write transfers.

Static memory Write: Timing and parameters

This section shows static memory write timing diagrams and parameters.

External memory write transfer with zero wait states

This diagram shows a single external memory write transfer with minimum zero wait states (WAITWR=0). One wait state is added.

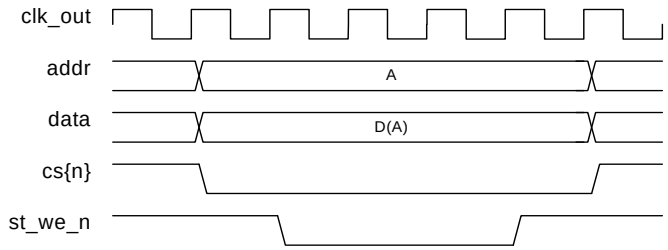


Downloaded from www.Manualslib.com manuals search engine

Timing parameters	Value
WAITRD	N/A
WAITOEN	N/A
WAITPAGE	N/A
WAITWR	0
WAITWEN	0
WAITTURN	N/A

External memory write transfer with two wait states

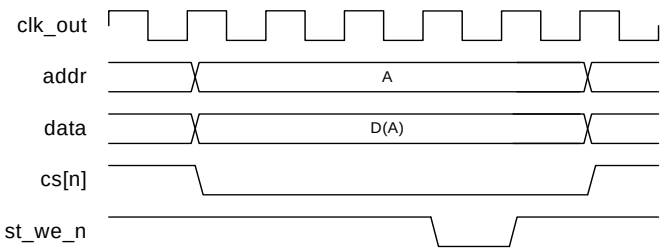
This diagram shows a single external memory write transfer with two wait states (WAITWR=2). One AHB wait state is added.



Timing parameter	Value
WAITRD	N/A
WAITOEN	N/A
WAITPAGE	N/A
WAITWR	2
WAITWEN	0
WAITTURN	N/A

External memory write transfer with two write enable delay states

This diagram shows a single external memory write transfer with two write enable delay states (WAITWEN=2). One wait state is added.

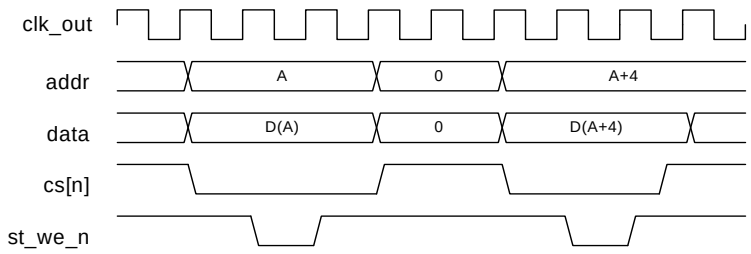


Downloaded from www.Manualslib.com manuals search engine

Timing parameters	Value
WAITRD	N/A
WAITOEN	N/A
WAITPAGE	N/A
WAITWR	2
WAITWEN	2
WAITTURN	N/A

Two external memory write transfers with zero wait states

This diagram shows two external memory write transfers with zero wait states (WAITWR=0). Four AHB wait states are added to the second write, because this write can be started only when the first write has completed. This is the timing of any sequence of write transfers, nonsequential to nonsequential or nonsequential to sequential, with any value of HBURST. The maximum speed of write transfers is controlled by the external timing of the write enable relative to the chip select, so all external writes must take two cycles to complete: the cycle in which write enable is asserted and the cycle in which write enable is deasserted.



Timing parameter	Value
WAITRD	N/A
WAITOEN	N/A
WAITPAGE	N/A
WAITWR	0
WAITWEN	0
WAITTURN	0

Flash memory

Write timing for flash memory is the same as for SRAM devices.

Downloaded from www.Manualslib.com manuals search engine

Bus turnaround

The memory controller can be configured for each memory bank to use external bus turnaround cycles between read and write memory accesses. The WAITTURN field can be programmed for 1 to 16 turnaround wait states, to avoid bus contention on the external memory databus. Bus turnaround cycles are generated between external bus transfers as follows:

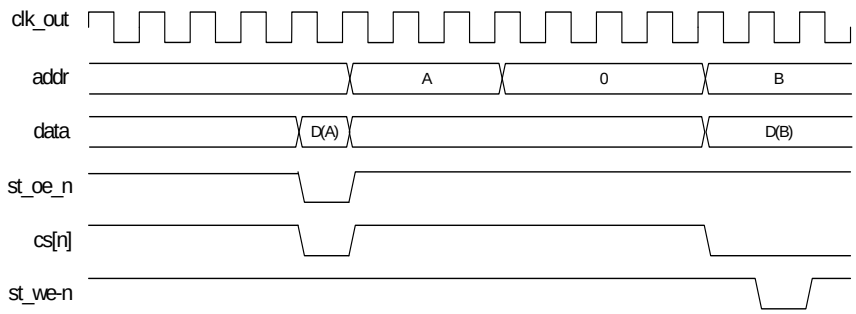
- Read to read (different memory banks)
- Read to write (same memory bank)
- Read to write (different memory banks)

Bus turnaround: Timing and parameters

This section shows bus turnaround timing diagrams and parameters.

Read followed by write with no turnaround

This diagram shows a zero wait read followed by a zero wait write with default turnaround between the transfers of two cycles because of the timing of the AHB transfers. Standard AHB wait states are added to the transfers, five for the read and three for the write.

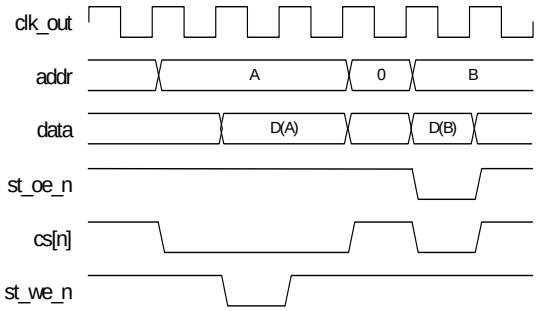


Timing parameter	Value
WAITRD	0
WAITOEN	0
WAITPAGE	N/A
WAITWR	0
WAITWEN	0
WAITTURN	0

Downloaded from www.Manualslib.com manuals search engine

Write followed by a read with no turnaround

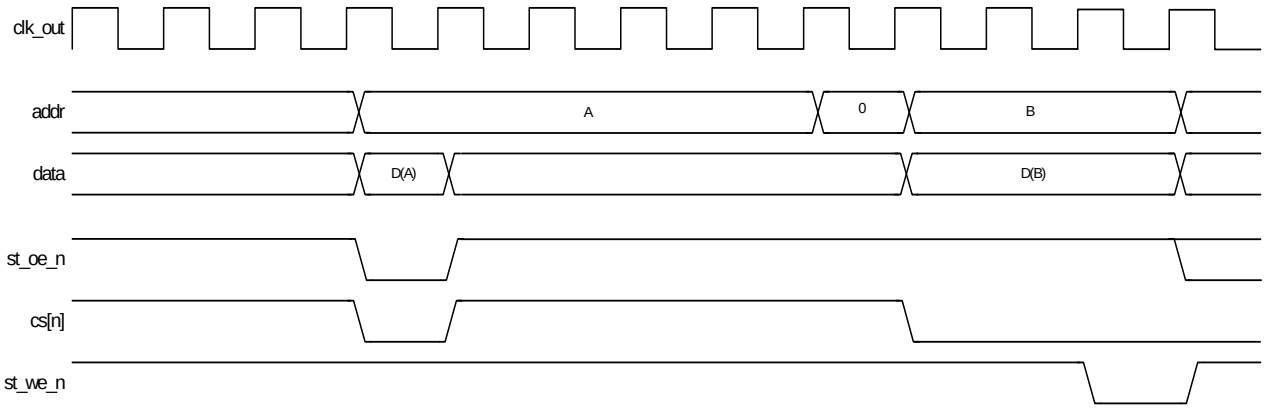
This diagram shows a zero wait write followed by a zero wait read with default turnaround between the transfers of one cycle. Three wait states are added to the write transfer; five wait states are added to the read transfer. The five AHB arbitration cycles for the read transfer include two wait states to allow the previous write access to complete and the three standard wait states for the read transfer.



Timing parameter	Value
WAITRD	0
WAITOEN	0
WAITPAGE	N/A
WAITWR	0
WAITWEN	0
WAITTURN	0

Read followed by a write with two turnaround cycles

TIs diagram shows a zero wait read followed by a zero wait write with two turnaround cycles added. The standard minimum of three AHB arbitration cycles is added to the read transfer and two wait states are added to the write transfer (as for any read-write transfer sequence).



Downloaded from www.Manualslib.com manuals search engine

Timing parameters	Value
WAITRD	0
WAITOEN	0
WAITPAGE	N/A
WAITWR	0
WAITWEN	0
WAITTURN	2

Byte lane control

The memory controller generates the byte lane control signals data_mask[3:0] according to these attributes:

- Little or big endian operation
- Transfer width
- External memory bank databus width, defined within each control register
- The decoded address value for write accesses only

Word transfers are the largest size transfers supported by the memory controller. Any access tried with a size greater that a word causes an error response. Each memory chip select can be 8, 16, or 32 bits wide. The memory type used determines how the st_we_n and data_mask signals are connected to provide byte, halfword, and word access.

For read accesses, you must control the data_mask signals by driving them all high or all low. Do this by programming the byte lane state (PB) bit in the Static Configuration [3:0] register. See “Address connectivity” on page 222 for additional information, with respect to st_we_n and data_mask, for different memory configurations.

Downloaded from www.Manualslib.com manuals search engine

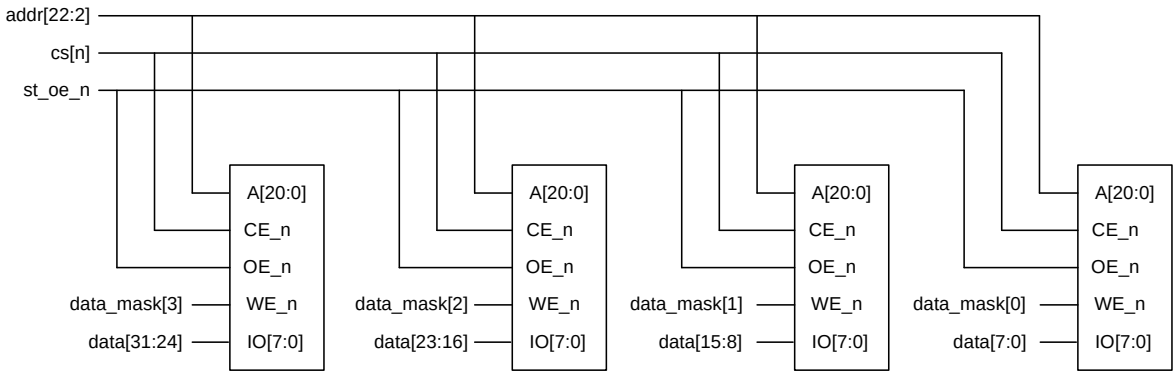
Address connectivity

Memory banks constructed from 8-bit or non-byte-partitioned memory devices

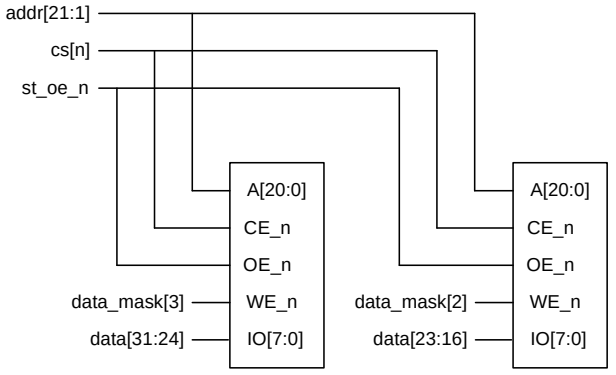
For memory banks constructed from 8-bit or non-byte-partitioned memory devices, it is important that the byte lane state (PB) bit is cleared to 0 within the respective memory bank control register. This forces all data_mask lines high during a read access, as the byte lane selects are connected to the device write enables.

The next figure shows 8-bit memory configuring memory banks that are 8-, 16-, and 32-bits wide. In each of these configurations, the data_mask[3:0] signals are connected to write enable (WE_n) inputs of each 8-bit memory. The st_we_n signal from the memory controller is not used.

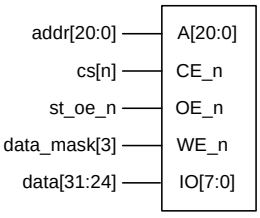
- For write transfers, the appropriate data_mask[3:0] byte lane signals are asserted low, and direct the data to the addressed bytes.
- For read transfers, all data_mask[3:0] signals are deasserted high, enabling the external bus to be defined for at least the width of the accessed memory.



32-bit bank consisting of four 8-bit devices



16-bit bank consisting of two 8-bit devices



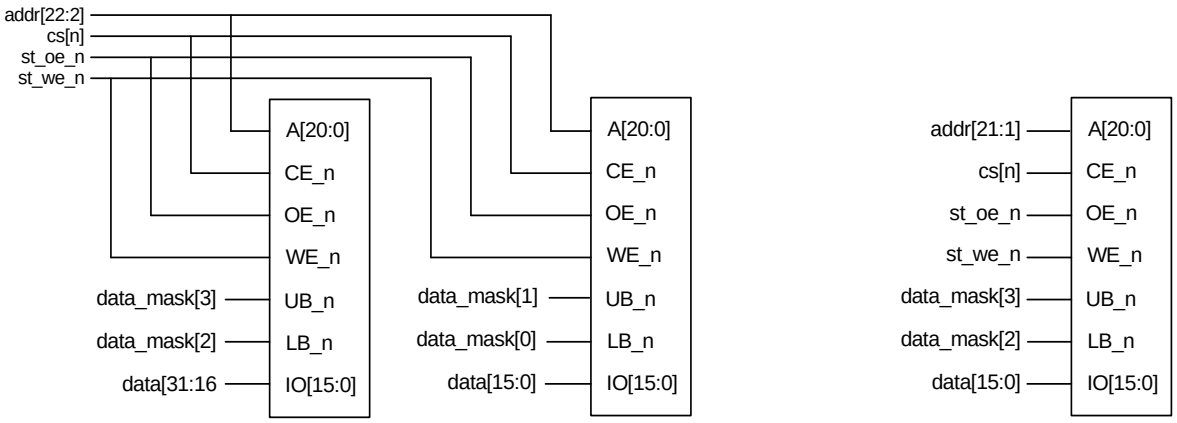
8-bit bank consisting of one 8-bit device

Downloaded from www.Manualslib.com manuals search engine

Memory banks constructed from 16-or 32-bit memory devices

For memory banks constructed from 16- or 32-bit memory devices, it is important that the byte lane select (PB) bit is set to 1 within the respective memory bank control register. This asserts all data_mask[3:0] lines low during a read access as, during a read, all device bytes must be selected to avoid undriven byte lanes on the read data value. With 16- and 32-bit wide memory devices, byte select signals exist and must be appropriately controlled; see the next two figures.

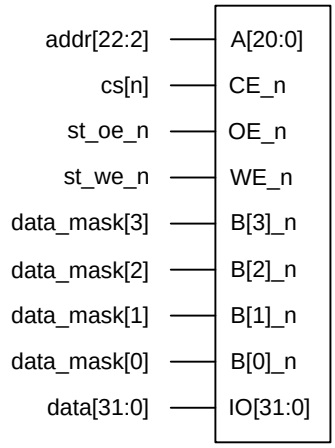
Memory banks constructed from 16-bit memory



32-bit bank consisting of two 16-bit devices

16-bit bank consisting of one 16-bit device

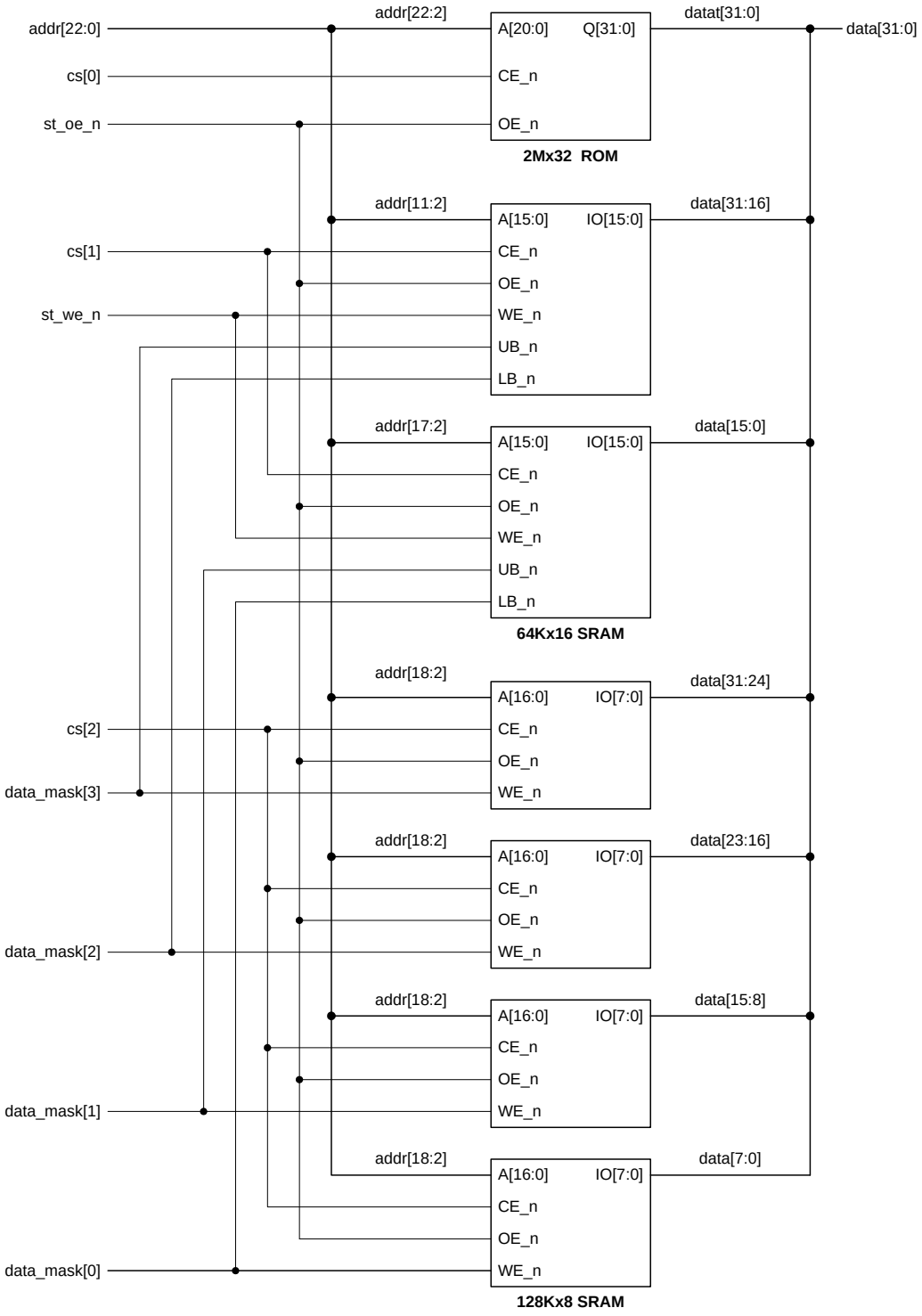
Memory bank constructed from 32-bit memory



32-bit bank consisting of one 32-bit device

The next figure shows connections for a typical memory system with different data width memory devices.

Downloaded from www.Manualslib.com manuals search engine



Downloaded from www.Manualslib.com manuals search engine

Dynamic memory controller

Write protection

Each dynamic memory chip select can be configured for write-protection by setting the appropriate bit in the write protect (P) field on the Dynamic Memory Configuration register. If a write access is performed to a write-protected memory bank, a bus error is generated.

Access sequencing and memory width

The data width of each chip select must be configured by programming the appropriate Dynamic Memory Configuration register. When the chip select data bus width is narrower than the transfer initiated from the current bus master, the internal bus transfer takes several external bus transfers to complete. If chip select 4 is configured as 16-bit wide memory, for example, and a 32-bit read is initiated, the AHB bus stalls while the memory controller reads two consecutive words from memory. During these accesses, the memory controller block demultiplexes the two 16-bit words into one 32-bit word and places the result onto the AHB bus.

Word transfers are the widest transfers supported by the memory controller. Any access tried with a size larger than a word generates an error response.

SDRAM Initialization

These steps show how to initialize an external SDRAM device:

1. Wait 100 ms after powerup and clocks have stabilized.
2. Set the SDRAMInit value in the Dynamic Control register to 11 — Issue SDRAM NOP command.
3. Wait 200 ms.
4. Set the SDRAMInit value in the Dynamic Control register to 10 — Issue SDRAM PALL (precharge all) command. This precharges all banks and places the SDRAM device into the *all banks idle* state.
5. Force frequent refresh cycles by writing a 1 to the Dynamic Refresh register. This provides a memory refresh every 16 memory clock cycles.
6. Wait until eight SDRAM refresh cycles have occurred (128 memory clock cycles).
7. Program the appropriate operational value to the Dynamic Refresh register.
8. Program the appropriate operational value to the Dynamic Ras and Cas *N* register.
9. Program the appropriate operational value to the Dynamic Configuration *N* register, with the exception of the buffer enable bit, which must be set to 0 during initialization.

Downloaded from www.Manualslib.com manuals search engine

10 Set the SDRAMInit value in the Dynamic Control register to 01 — Issue SDRAM Mode command.

11 Program the SDRAM memory 10-bit mode register. The mode register enables these parameters to be programmed:

Bit	Parameter	Parameter description
02:00	Burst length	■ 4 for a 32-bit wide external bus ■ 8 for a 16-bit wide external bus
03	Burst type	Sequential
06:04	CAS latency	Dependent on the SDRAM device and operating frequency
08:07	Operating mode	Standard operation
09	Write burst mode	Programmed burst length

A read transaction from the SDRAM memory programs the mode register. The transfer address contains the value to be programmed. Address bits 31:28 determine the chip select of the specific SDRAM that is being programmed. The 10-bit mode value must be shifted left per the specific device being programmed; see the tables following this procedure to determine the left shift value.

All other address bits must be set to 0.

12 Set the SDRAMInit value in the Dynamic Control register to 00 — Issue SDRAM normal operation command.

13 Enable the buffers by writing a 1 to the buffer enable bit in the Dynamic Configuration *N* register.

The SDRAM is now ready for normal operation.

Left-shift value table: 32-bit wide data bus SDRAM (RBC)

Device size	Configuration	Load Mode register left shift
16M	2 x 1M x 16	11
	4 x 2M x 8	12
64M	1 x 2M x 32	12
	2 x 4M x 16	12
	4 x 8M x 8	13
128M	1 x 4M x 32	12
	2 x 8M x 16	13
	4 x 16M x 8	14

Downloaded from www.Manualslib.com manuals search engine

Device size	Configuration	Load Mode register left shift
256M	1 x 8M x 32	12
	2 x 16M x 16	13
	4 x 32M x 8	14
512M	2 x 32M x 16	14
	4 x 64M x 8	15

Left-shift value table: 32-bit wide data bus SDRAM (BRC)

Device size	Configuration	Load Mode register left shift
16M	2 x 1M x 16	10
	4 x 2M x 8	11
64M	1 x 2M x 32	10
	2 x 4M x 16	10
	4 x 8M x 8	11
128M	1 x 4M x 32	10
	2 x 8M x 16	11
	4 x 16M x 8	12
256M	1 x 8M x 32	11
	2 x 16M x 16	11
	4 x 32M x 8	12
512M	2 x 32M x 16	12
	4 x 64M x 8	13

Left-shift value table: 16-bit wide data bus SDRAM (RBC)

Device size	Configuration	Load Mode register left shift
16M	1 x 1M x 16	10
	2 x 2M x 8	12
64M	1 x 4M x 16	11
	2 x 8M x 8	12
128	1 x 8M x 16	12
	2 x 16M x 8	13
256M	1 x 16M x 16	12
	2 x 32M x 8	13
512M	1 x 32M x 16	13
	2 x 64M x 8	14

Downloaded from www.Manualslib.com manuals search engine

Left-shift value table: 16-bit wide data bus SDRAM (BRC)

Device size	Configuration	Load Mode register left shift
16M	1 x 1M x 16	9
	2 x 2M x 8	10
64M	1 x 4M x 16	9
	2 x 8M x 8	10
128	1 x 8M x 16	10
	2 x 16M x 8	11
256M	1 x 16M x 16	10
	2 x 32M x 8	11
512M	1 x 32M x 16	11
	2 x 64M x 8	12

SDRAM address and data bus interconnect

The processor ASIC can connect to standard 16M and larger SDRAM components in either 16- or 32-bit wide configurations. The next tables show address and data bus connectivity. Note that for the 16-bit wide configuration the data bus connects to data [31:16] on the processor.

32-bit wide configuration

Signal	16M device SDRAM signal	64M device SDRAM signal	128M device SDRAM signal	256M device SDRAM signal	512M device SDRAM signal
addr[2]	A0	A0	A0	A0	A0
addr[3]	A1	A1	A1	A1	A1
addr[4]	A2	A2	A2	A2	A2
addr[5]	A3	A3	A3	A3	A3
addr[6]	A4	A4	A4	A4	A4
addr[7]	A5	A5	A5	A5	A5
addr[8]	A6	A6	A6	A6	A6
addr[9]	A7	A7	A7	A7	A7
addr[10]	A8	A8	A8	A8	A8
addr[11]	A9	A9	A9	A9	A9
addr[12]					
addr[13]		A11	A11	A11	A11

Downloaded from www.Manualslib.com manuals search engine

Signal	16M device SDRAM signal	64M device SDRAM signal	128M device SDRAM signal	256M device SDRAM signal	512M device SDRAM signal
addr[14]			A12*	A12	A12
addr[15]					
addr[16]					
addr[17]					
addr[18]					
addr[19]					
addr[20]					
addr[21]	BA				
addr[22]		BA0	BA0	BA0	BA0
addr[23]		BA1	BA1	BA1	BA1
ap10		A10/AP	A10/AP	A10/AP	A10/AP
data[31:0]		D[31:0]	D[31:0]	D[31:0]	D[31:0]
* A12 used only in 4 x 16M x 8 configurations					

32-bit wide configuration

Signal	16M device SDRAM signal	64M device SDRAM signal	128M device SDRAM signal	256M device SDRAM signal	512M device SDRAM signal
addr[1]	A0	A0	A0	A0	A0
addr[2]	A1	A1	A1	A1	A1
addr[3]	A2	A2	A2	A2	A2
addr[4]	A3	A3	A3	A3	A3
addr[5]	A4	A4	A4	A4	A4
addr[6]	A5	A5	A5	A5	A5
addr[7]	A6	A6	A6	A6	A6
addr[8]	A7	A7	A7	A7	A7
addr[9]	A8	A8	A8	A8	A8
addr[10]	A9	A9	A9	A9	A9
addr[11]					
addr[12]		A11	A11	A11	A11
addr[13]			A12*	A12	A12
addr[14]					

Downloaded from www.Manualslib.com manuals search engine

Signal	16M device SDRAM signal	64M device SDRAM signal	128M device SDRAM signal	256M device SDRAM signal	512M device SDRAM signal
addr[15]					
addr[16]					
addr[17]					
addr[18]					
addr[19]					
addr[20]	BA				
addr[21]		BA0	BA0	BA0	BA0
addr[22]		BA1	BA1	BA1	BA1
ap10		A10/AP	A10/AP	A10/AP	A10/AP
data[31:16]		D[15:0]	D[15:0]	D[15:0]	D[15:0]
* A12 used only in 2 x 16M x 8 configurations					

Registers

Register map

All configuration registers must be accessed as 32-bit words and as single accesses only. Bursting is not allowed.

Address	Register	Description
A070 0000	Control register	Control register
A070 0004	Status register	Status register
A070 0008	Config register	Configuration register
A070 0020	DynamicControl	Dynamic Memory Control register
A070 0024	DynamicRefresh	Dynamic Memory Refresh Timer
A070 0028	DynamicReadConfig	Dynamic Memory Read Configuration register
A070 0030	DynamictRP	Dynamic Memory Precharge Command Period (t_{RP})
A070 0034	DynamictRAS	Dynamic Memory Active to Precharge Command Period (t_{RAS})
A070 0038	DynamictSREX	Dynamic Memory Self-Refresh Exit Time (t_{SREX})
A070 003C	DynamictAPR	Dynamic Memory Last Data Out to Active Time (t_{APR})
A070 0040	DynamictDAL	Dynamic Memory Data-in to Active Command Time (t_{DAL} or T_{APW})

Downloaded from www.Manualslib.com manuals search engine

Address	Register	Description
A070 0044	DynamictWR	Dynamic Memory Write Recovery Time (t_{WR}, t_{DPL}, t_{RWL}, t_{RDL})
A070 0048	DynamictRC	Dynamic Memory Active to Active Command Period (t_{RC})
A070 004C	DynamictRFC	Dynamic Memory Auto Refresh Period, and Auto Refresh to Active Command Period (t_{RFC})
A070 0050	DynamictXSR	Dynamic Memory Exit Self-Refresh to Active Command (t_{XSR})
A070 0054	DynamictRRD	Dynamic Memory Active Bank A to Active B Time (t_{RRD})
A070 0058	DynamictMRD	Dynamic Memory Load Mode register to Active Command Time (t_{MRD})
A070 0080	StaticExtendedWait	Static Memory Extended Wait
A070 0100	DynamicConfig0	Dynamic Memory Configuration Register 0
A070 0104	DynamicRasCas0	Dynamic Memory RAS and CAS Delay 0
A070 0120	DynamicConfig1	Dynamic Memory Configuration Register 1
A070 0124	DynamicRasCas1	Dynamic Memory RAS and CAS Delay 1
A070 0140	DynamicConfig2	Dynamic Memory Configuration Register 2
A070 0144	DynamicRasCas2	Dynamic Memory RAS and CAS Delay 2
A070 0160	DynamicConfig3	Dynamic Memory Configuration Register 3
A070 0164	DynamicRasCas3	Dynamic Memory RAS and CAS Delay 3
A070 0200	StaticConfig0	Static Memory Configuration Register 0
A070 0204	StaticWaitWen0	Static Memory Write Enable Delay 0
A070 0208	StaticWaitOen0	Static Memory Output Enable Delay 0
A070 020C	StaticWaitRd0	Static Memory Read Delay 0
A070 0210	StaticWaitPage0	Static Memory Page Mode Read Delay 0
A070 0214	StaticWaitWr0	Static Memory Write Delay 0
A070 0218	StaticWaitTurn0	Static Memory Turn Round Delay 0
A070 0220	StaticConfig1	Static Memory Configuration Register 1
A070 0224	StaticWaitWen1	Static Memory Write Enable Delay 1
A070 0228	StaticWaitOen1	Static Memory Output Enable Delay 1
A070 022C	StaticWaitRd1	Static Memory Read Delay 1
A070 0230	StaticWaitPage1	Static Memory Page Mode Read Delay 1
A070 0234	StaticWaitWr1	Static Memory Write Delay 1
A070 0238	StaticWaitTurn1	Static Memory Turn Round Delay 1

Downloaded from www.Manualslib.com manuals search engine

Address	Register	Description
A070 0240	StaticConfig2	Static Memory Configuration Register 2
A070 0244	StaticWaitWen2	Static Memory Write Enable Delay 2
A070 0248	StaticWaitOen2	Static Memory Output Enable Delay 2
A070 024C	StaticWaitRd2	Static Memory Read Delay 2
A070 0250	StaticWaitPage2	Static Memory Page Mode Read Delay 2
A070 0254	StaticWaitWr2	Static Memory Write Delay 2
A070 0258	StaticWaitTurn2	Static Memory Turn Round Delay 2
A070 0260	StaticConfig3	Static Memory Configuration Register 3
A070 0264	StaticWaitWen3	Static Memory Write Enable Delay 3
A070 0268	StaticWaitOen3	Static Memory Output Enable Delay 3
A070 026C	StaticWaitRd3	Static memory Read Delay 3
A070 0270	StaticWaitPage3	Static Memory Page Mode Read Delay 3
A070 0274	StaticWaitWr3	Static Memory Write Delay 3
A070 0278	StaticWaitTurn3	Static Memory Turn Round Delay 3

Reset values

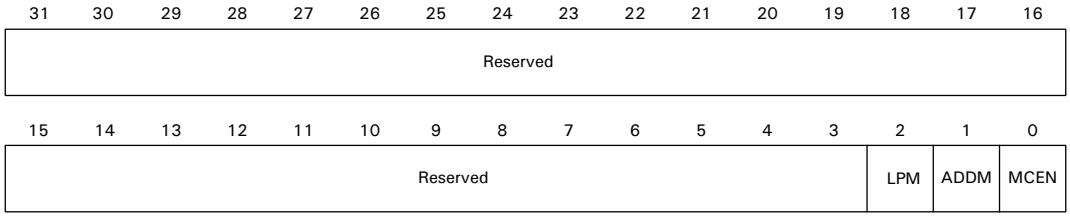
Reset values will be noted in the description column of each register table, rather than as a separate column.

Control register

Address: A070 0000

The Control register controls the memory controller operation. The control bits can be changed during normal operation.

Register



Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

Bits	Access	Mnemonic	Description
D31:03	N/A	Reserved	N/A (do not modify)
D02	R/W	LPM	**Low-power mode** 0 Normal mode (reset value on reset_n) 1 Low-power mode Indicates normal or low-power mode. Entering low-power mode reduces memory controller power consumption. Dynamic memory is refreshed as necessary. The memory controller returns to normal functional mode by clearing the low-power mode bit or by power-on reset. If you modify this bit, be sure the memory controller is in idle state. If you modify the L bit, be aware of these conditions: ■ The external memory cannot be accessed in low-power or disabled state. If a memory access is performed in either of these states, an error response is generated. ■ The memory controller AHB programming port can be accessed normally. ■ The memory controller registers can be programmed in low-power and/or disabled state.
D01	R/W	ADDM	**Address mirror** 0 Normal memory map 1 Reset memory map. Static memory chip select 1 is mirrored onto chip select 0 and chip select 4 (reset value on reset_n) Indicates normal or reset memory map. On power-on reset, chip select 1 is mirrored to both chip select 0 and chip select 1/chip select 4 memory areas. Clearing the M bit allows chip select 0 and chip select 4 memory to be accessed.
D00	R/W	MCEN	**Memory controller enable** 0 Disabled 1 Enabled (reset value on reset_n) Disabling the memory controller reduces power consumption. When the memory controller is disabled, the memory is not refreshed. The memory controller is enabled by setting the enable bit or by power-on reset. If you modify this bit, be sure the memory controller is in idle state. If you modify the E bit, be aware of these conditions: ■ The external memory cannot be accessed in low-power or disabled state. If a memory access is performed in either of these states, an error response is generated. ■ The memory controller AHB programming port can be accessed normally. ■ The memory controller registers can be programmed in low-power and/or disabled state.

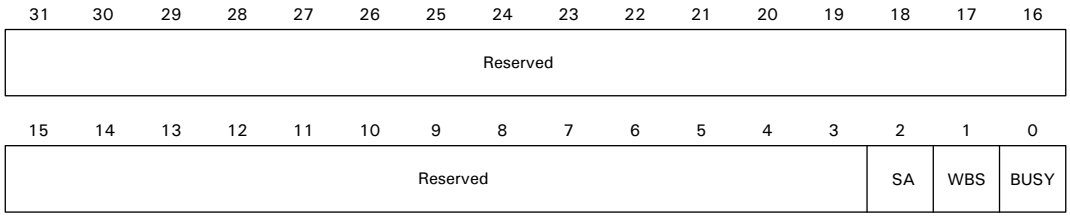
Downloaded from www.Manualslib.com manuals search engine

Status register

Address: A070 0004

The Status register provides memory controller status information.

Register



Register bit assignment

Bits	Access	Mnemonic	Description
D31:03	N/A	Reserved	N/A (do not modify)
D02	R	SA	**Self-refresh acknowledge (SREFACK)** 0 Normal mode 1 Self refresh mode (reset value on reset_n) Indicates the memory controller operating mode.
D01	R	WBS	**Write buffer status** 0 Write buffers empty (reset value on reset_n) 1 Write buffers contain data Enables the memory controller to enter low-power mode or disabled mode clearly.
D00	R	BUSY	**Busy** 0 Memory controller is idle 1 Memory controller is busy performing memory transactions, commands, or auto-refresh cycles, or is in self-refresh mode (reset value on reset_n). Ensures that the memory controller enters the low-power or disabled state cleanly by determining whether the memory controller is busy.

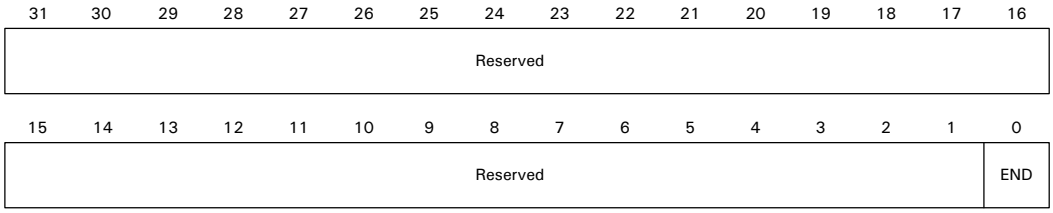
Configuration register

Address: A070 0008

The Configuration register configures memory controller operation. It is recommended that this register be modified during system initialization, or when there are no current or outstanding transactions. Wait until the memory controller is idle, then enter low-power or disabled mode.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

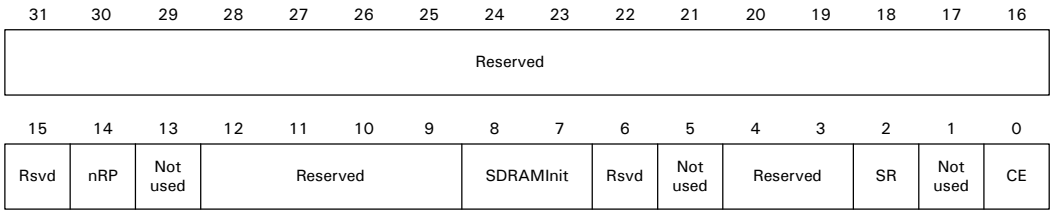
Bits	Access	Mnemonic	Description
D31:01	N/A	Reserved	N/A (do not modify)
D00	R/W	END	Endian mode 0 Little endian mode 1 Big endian mode The value of the endian bit on power-on reset (reset_n) is determined by the gpio_a[3] signal. This value can be overridden by software. Note: The value of the gpio_a[3] signal is reflected in this field. When programmed, this register reflects the last value written into the register. You must flush all data in the memory controller before switching between little endian and big endian modes.

Dynamic Memory Control register

Address: A070 0020

The Dynamic Memory Control register controls dynamic memory operation. The control bits can be changed during normal operation.

Register



Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

Bits	Access	Mnemonic	Description
D31:15	N/A	Reserved	N/A (do not modify)
D14	R/W	nRP	**Sync/Flash reset/power down signal (dy_pwr_n)** 0 dy_pwr_n signal low (reset value on reset_n) 1 Set dy_pwr_n signal high
D13	R/W	Not used	Always write to 0.
D12:09	N/A	Reserved	N/A (do not modify)
D08:07	R/W	SDRAMInit	**SDRAM initialization** 00 Issue SDRAM NORMAL operation command (reset value on reset_n) 01 Issue SDRAM MODE command 10 Issue SDRAM PALL (precharge all) command 11 Issue SDRAM NOP (no operation) command
D06	N/A	Reserved	N/A (do not modify)
D05	R/W	Not used	Must write 0.
D04:03	N/A	Reserved	N/A (do not modify)
D02	R/W	SR	**Self-refresh request (SREFREQ)** 0 Normal mode 1 Enter self-refresh mode (reset value on reset_n) By writing 1 to this bit, self-refresh can be entered under software control. Writing 0 to this bit returns the memory controller to normal mode. The self-refresh acknowledge bit in the Status register must be polled to discover the current operating mode of the memory controller. Note: The memory controller exits from power-on reset with the self-refresh bit on high. To enter normal functional mode, set the self-refresh bit low. Writing to this register with the bit set to high places the register into self-refresh mode. This functionality allows data to be stored over SDRAM self-refresh of the ASIC is powered down.
D01	R/W	Not used	Must write 1.
D00	R/W	CE	**Dynamic memory clock enable** 0 Clock enable if idle devices are deasserted to save power (reset value on reset_n) 1 All clock enables are driven high continuously. Note: Clock enable must be high during SDRAM initialization.

Dynamic Memory Refresh Timer register

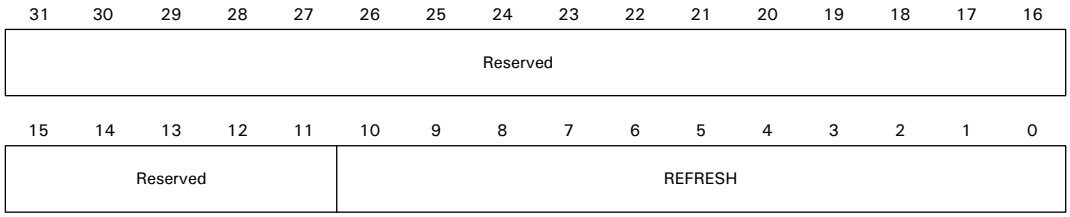
Address: A070 0024

Downloaded from www.Manualslib.com manuals search engine

The Dynamic Memory Refresh Timer register configures dynamic memory operation. It is recommended that this register be modified during system initialization, or when there are no current or outstanding transactions. Wait until the memory controller is idle, then enter low-power or disabled mode.These bits can, however, be changed during normal operation if necessary.

Note: The Dynamic Memory Refresh Timer register is used for all four dynamic memory chip selects. The worst case value for all chip selects must be programmed.

Register



Register bit assignment

Bits	Access	Mnemonic	Description
D31:11	N/A	Reserved	N/A (do not modify)
D10:00	R/W	REFRESH	**Refresh timer** 0x0 Refresh disabled (reset value on reset_n) 0x1–0x77F n(x16) 16n clk_out ticks between SDRAM refresh cycles

Note: The refresh cycles are evenly distributed. There might be slight variations, however, when the auto-refresh command is issued, depending on the status of the memory controller.

Dynamic Memory Read Configuration register

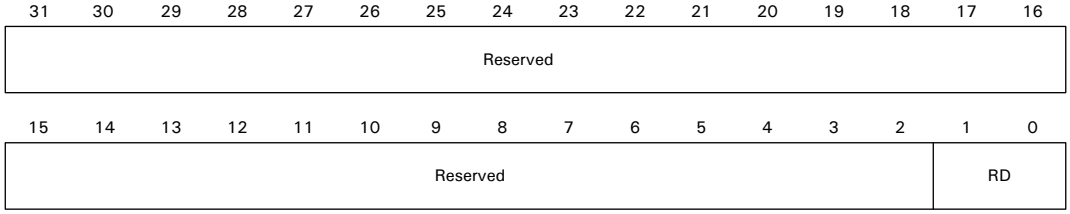
Address: A070 0028

The Dynamic Memory Read Configuration register allows you to configure the dynamic memory read strategy. Modify this register only during system initialization.

Note: The Dynamic Memory Read Configuration register is used for all four dynamic memory chip selects. The worst case value for all chip selects must be programmed.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

Bits	Access	Mnemonic	Description
D31:02	N/A	Reserved	N/A (do not modify)
D01:00	R/W	RD	**Read data strategy** 00 Reserved. 01 Command delayed strategy, using CLKDELAY (command delayed, clock out not delayed). 10 Command delayed strategy plus one clock cycle, using CLKDELAY (command delayed, clock out not delayed). 11 Command delayed strategy plus two clock cycles, using CLKDELAY (command delayed, clock out not delayed).

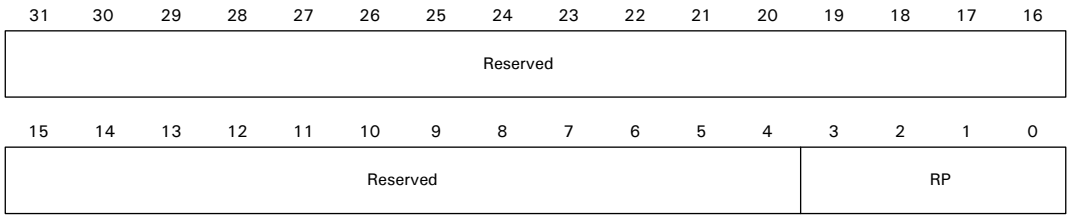
Dynamic Memory Precharge Command Period register

Address: A070 0030

The Dynamic Memory Precharge Command Period register allows you to program the precharge command period, t_{RP}. Modify this register only during system initialization. This value normally is found in SDRAM datasheets as t_{RP}.

Note: The Dynamic Memory Precharge Command Period register is used for all four dynamic memory chip selects. The worst case value for all chip selects must be programmed.

Register



Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

Bits	Access	Mnemonic	Description
D31:04	N/A	Reserved	N/A (do not modify)
D03:00	R/W	RP	**Precharge command period (t_{RP})** 0x0–0xE n+1 clock cycles, where the delay is in clk_out cycles. **0xF** 16 clock cycles (reset value on reset_n)

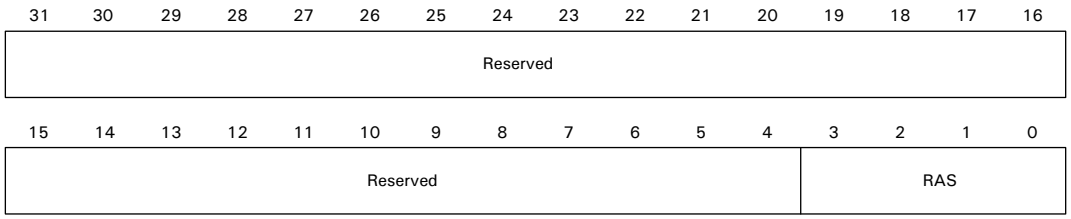
Dynamic Memory Active to Precharge Command Period register

Address: A070 0034

The Dynamic Memory Active to Precharge Command Period register allows you to program the active to precharge command period, t_{RAS}. It is recommended that this register be modified during system initialization, or when there are no current or outstanding transactions. Wait until the memory controller is idle, then enter low-power or disabled mode. This value normally is found in SDRAM datasheets as t_{RAS}.

Note: The Dynamic Memory Active to Precharge Command Period register is used for all four dynamic memory chip selects. The worst case value for all chip selects must be programmed.

Register



Register bit assignment

Bits	Access	Mnemonic	Description
D31:04	N/A	Reserved	N/A (do not modify)
D03:00	R/W	RAS	**Active to precharge command period (t_{RAS})** 0x0–0xE n+1 clock cycles, where the delay is in clk_out cycles. **0xF** 16 clock cycles (reset value on reset_n)

Downloaded from www.Manualslib.com manuals search engine

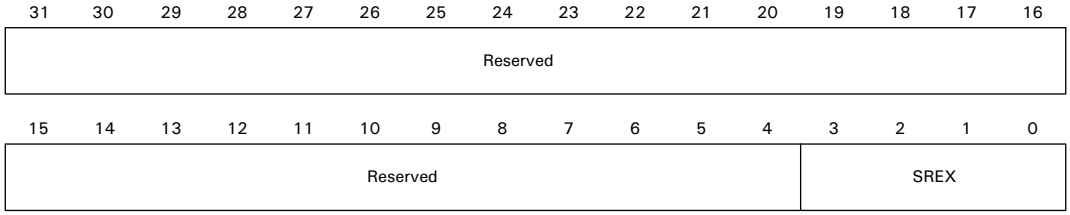
Dynamic Memory Self-refresh Exit Time register

Address: A070 0038

The Dynamic Memory Self-refresh Exit Time register allows you to program the self-refresh exit time, t_{SREX}. It is recommended that this register be modified during system initialization, or when there are no current or outstanding transactions. Wait until the memory controller is idle, then enter low-power or disabled mode. This value normally is found in SDRAM data sheets as t_{SREX}.

Note: The Dynamic Memory Self-refresh Exit Time register is used for all four dynamic memory chip selects. The worst case value for all chip selects must be programmed.

Register



Register bit assignment

Bits	Access	Mnemonic	Description
D31:04	N/A	Reserved	N/A (do not modify)
D03:00	R/W	SREX	**Self-refresh exit time (t_{SREX})** 0x0–0xE n+1 clock cycles, where the delay is in clk_out cycles. **0xF** 16 clock cycles (reset value on reset_n)

Dynamic Memory Last Data Out to Active Time register

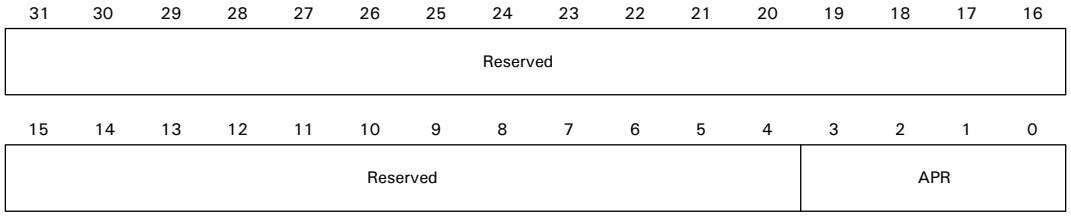
Address: A070 003C

The Dynamic Memory Last Data Out to Active Time register allows you to program the last-data-out to active command time, t_{APR}. It is recommended that this register be modified during system initialization, or when there are no current or outstanding transactions. Wait until the memory controller is idle, then enter low-power or disabled mode. This value normally is found in SDRAM datasheets as t_{APR}.

Note: The Dynamic Memory Last Data Out to Active Time register is used for all four dynamic memory chip selects. The worst case value for all chip selects must be programmed.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

Bits	Access	Mnemonic	Description
D31:04	N/A	Reserved	N/A (do not modify)
D03:00	R/W	APR	**Last-data-out to active command time (t_{APR})** 0x0–0xE n+1 clock cycles, where the delay is in clk_out cycles. **0xF** 16 clock cycles (reset value on reset_n)

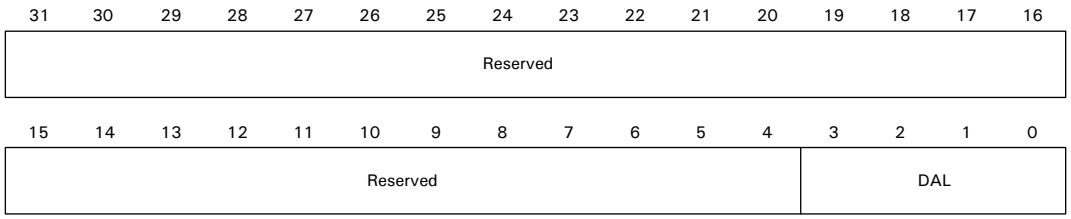
Dynamic Memory Data-in to Active Command Time register

Address: A070 0040

The Dynamic Memory Data-in to Active Command Time register allows you to program the data-in to active command time, t_{DAL}. It is recommended that this register be modified during system initialization, or when there are no current or outstanding transactions. Wait until the memory controller is idle, then enter low-power or disabled mode. This value normally is found in SDRAM data sheets as t_{DAL} or t_{APW}.

Note: The Dynamic Memory Data-in Active Command Time register is used for all four dynamic memory chip selects. The worst case value for all chip selects must be programmed.

Register



Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

Bits	Access	Mnemonic	Description
D31:04	N/A	Reserved	N/A (do not modify)
D03:00	R/W	DAL	**Data-in to active command (t_{DAL} or t_{APW})** 0x0–0xE n+1 clock cycles, where the delay is in clk_out cycles. **0xF** 15 clock cycles (reset value on reset_n)

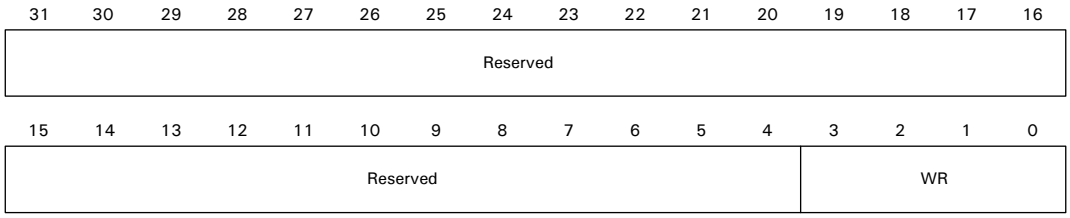
Dynamic Memory Write Recovery Time register

Address: A070 0044

The Dynamic Memory Write Recovery Time register allows you to program the write recovery time, t_{WR}. It is recommended that this register be modified during system initialization, or when there are no current or outstanding transactions. Wait until the memory controller is idle, then enter low-power or disabled mode. This value normally is found in SDRAM datasheets as t_{WR}, t_{DPL}, t_{RWL}, or t_{RDL}.

Note: The Dynamic Memory Write Recovery Time register is used for all four dynamic memory chip selects. The worst case value for all chip selects must be programmed.

Register



Register bit assignment

Bits	Access	Mnemonic	Description
D31:04	N/A	Reserved	N/A (do not modify)
D03:00	R/W	WR	Write recovery time (t_{WR}, t_{DPL}, t_{RWL}, or t_{RDL}) 0x0–0xE n+1 clock cycles, where the delay is in clk_out cycles. **0xF** 16 clock cycles (reset value on reset_n)

Downloaded from www.Manualslib.com manuals search engine

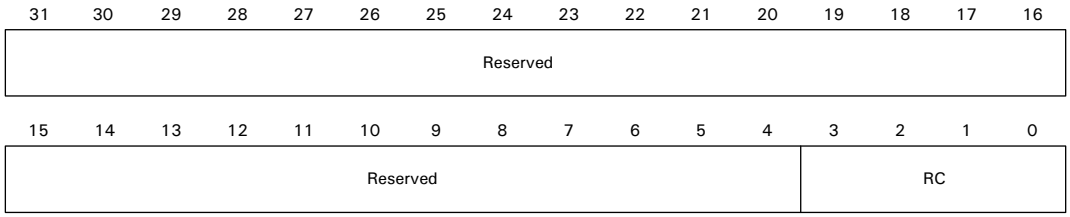
Dynamic Memory Active to Active Command Period register

Address: A070 0048

The Dynamic Memory Active to Active Command Period register allows you to program the active to active command period, t_{RC}. It is recommended that this register be modified during system initialization, or when there are no current or outstanding transactions. Wait until the memory controller is idle, then enter low-power or disabled mode. This value normally is found in SDRAM datasheets as t_{RC}.

Note: The Dynamic Memory Active to Active Command period register is used for all four dynamic memory chip selects. The worst case value for all chip selects must be programmed.

Register



Register bit assignment

Bits	Access	Mnemonic	Description
D31:05	N/A	Reserved	N/A (do not modify)
D04:00	R/W	RC	**Active to active command period (t_{RC})** 0x0–0x1E n+1 clock cycles, where the delay is in clk_out cycles. **0x1F** 32 clock cycles (reset value on reset_n)

Dynamic Memory Auto Refresh Period register

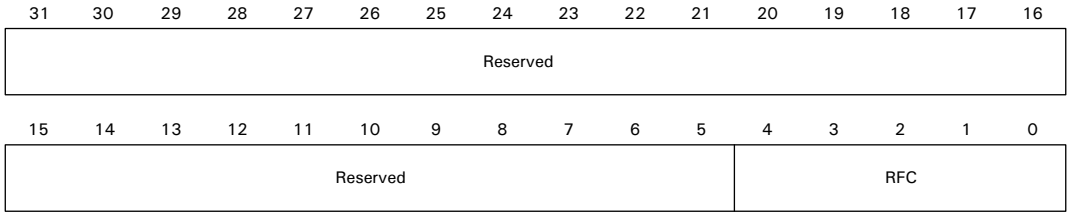
Address: A070 004C

The Dynamic Memory Auto Refresh Period register allows you to program the auto-refresh period and the auto-refresh to active command period, t_{RFC}. It is recommended that this register be modified during initialization, or when there are no current or outstanding transactions. Wait until the memory controller is idle, then enter low-power or disabled mode. This value normally is found in SDRAM datasheets as t_{RFC} or t_{RC}.

Downloaded from www.Manualslib.com manuals search engine

Note: The Dynamic Memory Auto Refresh Period register is used for all four dynamic memory chip selects. The worst case value for all chip selects must be programmed.

Register



Register bit assignment

Bits	Access	Mnemonic	Description
D31:05	N/A	Reserved	N/A (do not modify)
D04:00	R/W	RFC	**Auto-refresh period and auto-refresh to active command period** 0x0–0x1E n+1 clock cycles, where the delay is in clk_out cycles 0**x1F** 32 clock cycles (reset value on reset_n)

Dynamic Memory Exit Self-refresh register

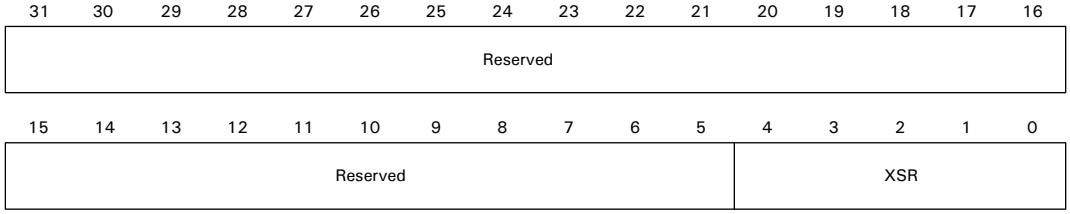
Address: A070 0050

The Dynamic Memory Exit Self-refresh register allows you to program the exit self-refresh to active command time, t_{XSR}. It is recommended that this register be modified during system initialization, or when there are no current or outstanding transactions. Wait until the memory controller is idle, then enter low-power or disabled mode. This value normally is found in SDRAM datasheets as t_{XSR}.

Note: The Dynamic Memory Exit Self-refresh register is used for all four dynamic memory chip selects. The worst case value for all the chip selects must be programmed.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

Bits	Access	Mnemonic	Description
D31:05	N/A	Reserved	N/A (do not modify)
D04:00	R/W	XSR	**Exit self-refresh to active time command** 0x0–0x1E n+1 clock cycles, where the delay is in clk_out cycles 0**x1F** 32 clock cycles (reset value on reset_n)

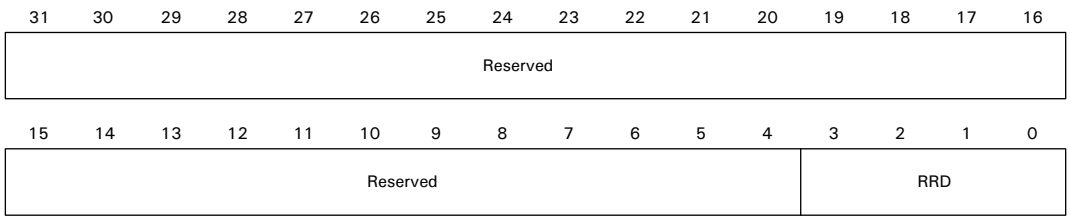
Dynamic Memory Active Bank A to Active Bank B Time register

Address: A070 0054

The Dynamic Memory Active Bank A to Active Bank B Time register allows you to program the active bank A to active bank B latency, t_{RRD}. It is recommended that this register be modified during system initialization, or when there are no current or outstanding transactions. Wait until the memory controller is idle, then enter low-power or disabled mode. This value normally is found in SDRAM datasheets as t_{RRD}.

Note: The Dynamic Memory Active Bank A to Active Bank B Time register is used for all four dynamic memory chip selects. The worst case value for all chip selects must be programmed.

Register



Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

Bits	Access	Mnemonic	Description
D31:04	N/A	Reserved	N/A (do not modify)
D03:00	R/W	RRD	**Active Bank A to Active Bank B** 0x0–0xE n+1 clock cycles, where the delay is in clk_out cycles 0**xF** 16 clock cycles (reset on reset_n)

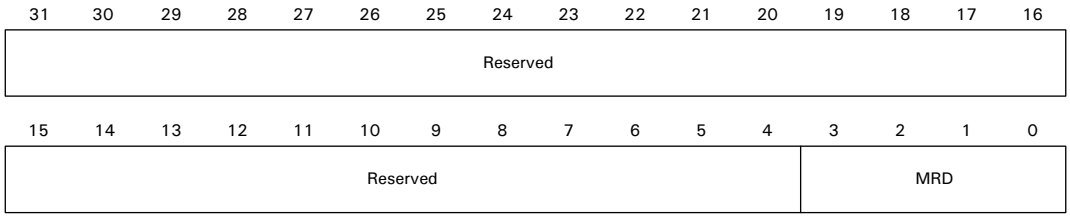
Dynamic Memory Load Mode register to Active Command Time register

Address: A070 0058

The Dynamic Memory Load Mode register to Active Command Time register allows you to program the Load Mode register to active command time, t_{MRD}. It is recommended that this register be modified during system initialization, or when there are no current or outstanding transactions. Wait until the memory controller is idle, then enter low-power or disabled mode. This value normally is found in SDRAM datasheets as t_{MRD} or t_{RSA}.

Note: The Dynamic Memory Load Mode register to Active Command Time register is used for all four chip selects. The worst case value for all chip selects must be programmed.

Register



Register bit assignment

Bits	Access	Mnemonic	Description
D31:045	N/A	Reserved	N/A (do not modify)
D03:00	R/W	MRD	**Load mode register to Active Command Time** 0x0–0xE n+1 clock cycles, where the delay is in clk_out cycles 0**xF** 16 clock cycles (reset on reset_n)

Downloaded from www.Manualslib.com manuals search engine

Static Memory Extended Wait register

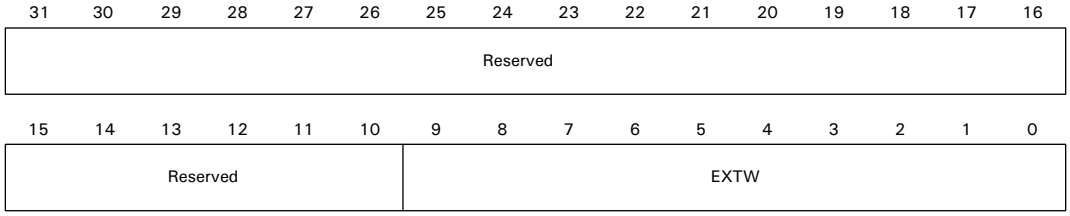
Address: A070 0080

The Static Memory Extended Wait register times long static memory read and write transfers (which are longer than can be supported by the Static Memory Read Delay registers or the Static Memory Write Delay registers) when the EW (extended wait) bit in the related Static Memory Configuration register is enabled.

There is only one Static Memory Extended Wait register, which is used by the relevant static memory chip select if the appropriate EW bit is set in the Static Memory Configuration register.

It is recommended that this register be modified during system initialization, or when there are no current or outstanding transactions. If necessary, however, these control bits can be changed during normal operation.

Register



Register bit assignment

Bits	Access	Mnemonic	Description
D31:10	N/A	Reserved	N/A (do not modify)
D09:00	R/W	EXTW	**External wait timeout** 0x0 16 clock cycles, where the delay is in clk_out cycles 0**x1-0x3FF** (n=1) x 16 clock cycles

Example

Static memory read/write time = 16 µs

CLK frequency = 50 MHz

This value must be programmed into the Static Memory Extended Wait register:

$(16 \times 10^{-6} \times 50 \times 10^{6} / 16) - 1 = 49$

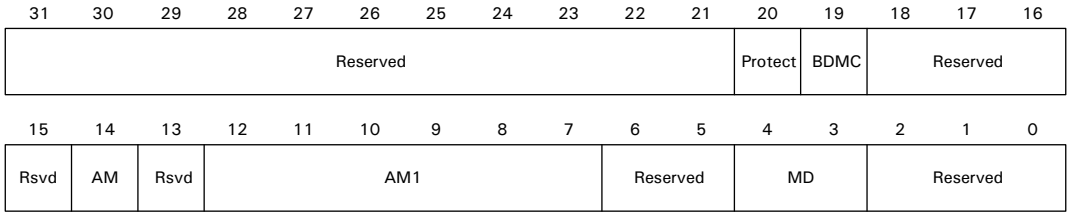
Dynamic Memory Configuration 0-3 registers

Address: A070 0100 / 0120 / 0140 / 0160

Downloaded from www.Manualslib.com manuals search engine

Use the Dynamic Memory Configuration 0–3 registers to program the configuration information for the relevant dynamic memory chip select. These registers are usually modified only during system initialization.

Register



Register bit assignment

Bits	Access	Mnemonic	Description
D31:21	N/A	Reserved	N/A (do not modify)
D20	R/W	Protect	**Write protect** 0 Writes not protected (reset value on reset_n) 1 Write protected
D19	R/W	BDMC	**Buffer enable** 0 Buffer disabled for accesses to this chip select (reset value on reset_n) 1 Buffer enabled for accesses to this chip select. The buffers must be disabled during SDRAM initialization. The buffers must be enabled during normal operation.
D18:15	N/A	Reserved	N/A (do not modify)
D14	R/W	AM	**Address mapping** 0 Reset value on reset_n See Table , “Register map,” on page 230 for more information.
D13	N/A	Reserved	N/A (do not modify)
D12:07	R/W	AM1	**Address mapping** 00000000 Reset value on reset_n The SDRAM column and row width and number of banks are computed automatically from the address mapping. See "Register map," beginning on page 230, for more information.
D06:05	N/A	Reserved	N/A (do not modify)
D04:03	R/W	MD	**Memory device** 00 SDRAM (reset value on reset_n) 01 Low-power SDRAM 10 Reserved 11 Reserved
D02:00	N/A	Reserved	N/A (do not modify)

Downloaded from www.Manualslib.com manuals search engine

Address mapping for the Dynamic Memory Configuration registers

The next table shows address mapping for the Dynamic Memory Configuration 0-3 registers. Address mappings that are not shown in the table are reserved.

[14]	[12]	[11:9]	[8:7]	Description
16-bit external bus high-performance address mapping (row, bank column)				
0	0	000	00	16 Mb (2Mx8), 2 banks, row length=11, column length=9
0	0	000	01	16 Mb (1Mx16), 2 banks, row length=11, column length=8
0	0	001	00	64 Mb (8Mx80, 4 banks, row length=12, column length=9
0	0	001	01	64 Mb (4Mx16), 4 banks, row length=12, column length=8
0	0	010	00	128 Mb (16Mx8), 4 banks, row length=12, column length=10
0	0	010	01	128 Mb (8Mx16), 4 banks, row length=12, column length=9
0	0	011	00	256 Mb (32Mx8), 4 banks, row length=13, column length=10
0	0	011	01	256 Mb (16Mx16), 4 banks, row length=13, column length=9
0	0	100	00	512 Mb (64Mx8), 4 banks, row length=13, column length=11
0	0	100	01	512 Mb (32Mx16), 4 banks, row length=13, column length=10
16-bit external bus low-power SDRAM address mapping (bank, row, column)				
0	1	000	00	16 Mb (2Mx8), 2 banks, row length=11, column length=9
0	1	000	01	16 Mb (1Mx16), 2 banks, row length=11, column length=8
0	1	001	00	64 Mb (8Mx8), 4 banks, row length 12, column length=9
0	1	001	01	64 Mb (4Mx16), 4 banks, row length=12, column length=8
0	1	010	00	128 Mb (16Mx8), 4 banks, row length=12, column length=10
0	1	010	01	128 Mb (8Mx16), 4 banks, row length=12, column length=9
0	1	011	00	256 Mb (32Mx8), 4 banks, row length=13, column length=10
0	1	011	01	256 Mb (16Mx16), 4 banks, row length=13, column length=9
0	1	100	00	512 Mb (64Mx8), 4 banks, row length=13, column length=11
0	1	100	01	512 Mb (32Mx16, 4 banks, row length=13, column length=10
32-bit extended bus high-performance address mapping (row, bank, column)				
1	0	000	00	16 Mb (2Mx8), 2 banks, row length=11, column length=9
1	0	000	01	16 Mb (1Mx16), 2 banks, row length=11, column length=8
1	0	001	00	64 Mb (8Mx8), 4 banks, row length=12, column length=9
1	0	001	01	64 Mb (4Mx16), 4 banks, row length=12, column length=8
1	0	001	10	64 Mb (2Mx32), 4 banks, row length=11, column length=8
1	0	010	00	128 Mb (16Mx8), 4 banks, row length=12, column length=10
1	0	010	01	128 Mb (8Mx16), 4 banks, row length=12, column length=9
1	0	010	10	128 Mb (4Mx32), 4 banks, row length=12, column length=8
1	0	011	00	256 Mb (32Mx8), 4 banks, row length=13, column length=10

Downloaded from www.Manualslib.com manuals search engine

[14]	[12]	[11:9]	[8:7]	Description
1	0	011	01	256 Mb (16Mx16), 4 banks, row length=13, column length=9
1	0	011	10	256 Mb (8Mx32), 4 banks, row length=13, column length=8
1	0	100	00	512 Mb (64Mx8), 4 banks, row length=13, column length=11
1	0	100	01	512 Mb (32Mx16), 4 banks, row length=13, column length=10
32-bit extended bus low-power SDRAM address mapping (bank, row, column)				
1	1	000	00	16 Mb (2Mx8), 2 banks, row length=11, column length=9
1	1	000	01	16 Mb (1Mx16), 2 banks, row length=11, column length=8
1	1	001	00	64 Mb (8Mx8), 4 banks, row length=12, column length=9
1	1	001	01	64 MB (4Mx16), 4 banks, row length=12, column length=8
1	1	001	10	64 Mb (2Mx32), 4 banks, row length=11, column length=8
1	1	010	00	128 Mb (16Mx8), 4 banks, row length=12, column length=10
1	1	010	01	128 Mb (8Mx16), 4 banks, row length=12, column length=9
1	1	010	10	128 Mb (4Mx32), 4 banks, row length=12, column length=8
1	1	011	00	256 Mb (32Mx8), 4 banks, row length=13, column length=10
1	1	011	01	256 Mb (16Mx16), 4 banks, row length=13, column length=9
1	1	011	10	256 Mb (8Mx32), 4 banks, row length=13, column length=8
1	1	100	00	512 Mb (64Mx8), 4 banks, row length=13, column length=11
1	1	100	01	512 Mb (32Mx16), 4 banks, row length=13, column length=10

Chip select and memory devices

A chip select can be connected to a single memory device; in this situation, the chip select data bus width is the same as the device width. As an alternative, the chip select can be connected to a number of external devices. In this situation, the chip select data bus width is the sum of the memory device databus widths.

Chip select and memory devices: Examples

For a chip select connected to	Select this mapping
32-bit wide memory device	32-bit wide address mapping
16-bit wide memory device	16-bit wide address mapping
4 x 8-bit wide memory devices	32-bit wide address mapping
2 x 8-bit memory devices	16-bit wide address mapping

Dynamic Memory RAS and CAS Delay 0-3 registers

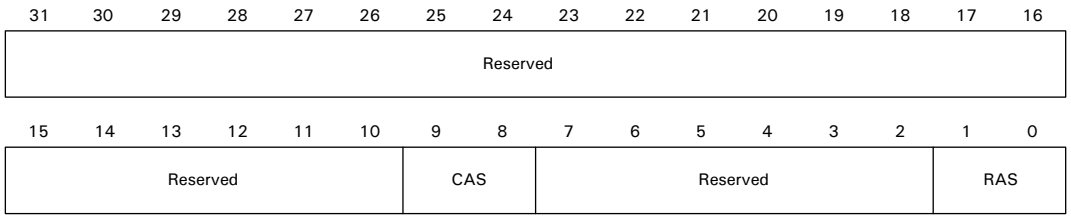
Address: A070 0104 / 0124 / 0144 / 0164

Downloaded from www.Manualslib.com manuals search engine

The Dynamic Memory RAS and CAS Delay 0–3 registers allow you to program the RAS and CAS latencies for the relevant dynamic memory. It is recommended that these registers be modified during system initialization, or when there are no current or outstanding transactions. Wait until the memory controller is idle, then enter low-power or disabled mode.

Note: The values programmed into these registers must be consistent with the values used to initialize the SDRAM memory device.

Register



Register bit assignment

Bits	Access	Mnemonic	Description
D31:10	N/A	Reserved	N/A (do not modify)
D09:08	R/W	CAS	**CAS latency** 00 Reserved 01 One clock cycle, where the RAS to CAS latency (RAS) and CAS latency (CAS) are defined in clk_out cycles 10 Two clock cycles 11 Three clock cycles (reset value on reset_n)
D07:02	N/A	Reserved	**N/A (do not modify)**
D01:00	R/W	RAS	**RAS latency (active to read/write delay)** 00 Reserved 01 One clock cycle, where the RAS to CAS latency (RAS) and CAS latency (CAS) are defined in clk_out cycles 10 Two clock cycles 11 Three clock cycles (reset value on reset_n)

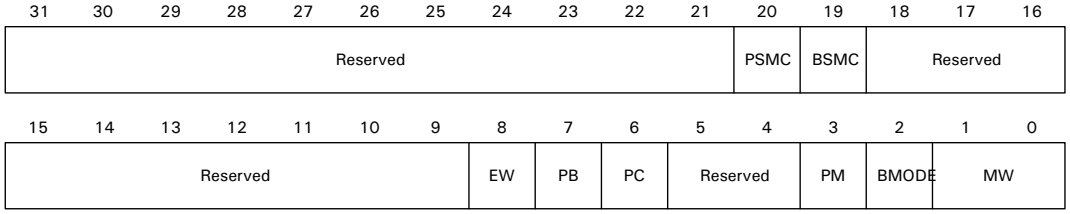
StaticMemory Configuration 0-3 registers

Address: A070 0200 / 0220 / 0240 / 0260

The Static Memory Configuration 0–3 registers configure the static memory configuration. It is recommended that these registers be modified during system initialization, or when there are no current or outstanding transactions. Wait until the memory controller is idle, then enter low-power or disabled mode.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

Bits	Access	Mnemonic	Description
D31:21	N/A	Reserved	N/A (do not modify)
D20	R/W	PSMC	**Write protect** 0 Writes not protected (reset value on reset_n) 1 Write protected
D19	R/W	BSMC	**Buffer enable** 0 Write buffer disabled (reset value on reset_n) 1 Write buffer enabled Note: This field must always be set to 0 when a peripheral other than SRAM is attached to the static ram chip select.
D18:09	N/A	Reserved	N/A (do not modify)
D08	R/W	EW	**Extended wait** 0 Extended wait disabled (reset value on reset_n) 1 Extended wait enabled Extended wait uses the Static Extended Wait register to time both the read and write transfers, rather than the Static Memory Read Delay 0–3 registers and Static Memory Write Delay 0–3 registers. This allows much longer transactions. Extended wait also can be used with the ns_ta_strb signal to allow a slow peripheral to terminate the access. In this case, the Static Memory Extended Wait register can be programmed with the maximum timeout limit. A high value on ns_ta_strb is then used to terminate the access before the maximum timeout occurs. Note: Extended wait and page mode cannot be selected simultaneously.

Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Description
D07	R/W	PB	**Byte lane state** 0 For reads, all bits in byte_lane[3:0] are high. For writes, the respective active bits in byte_lane[3:0] are low (reset value for chip select 0, 2, and 3 on reset_n). 1 For reads, the respective active bits in byte_lane[3:0] are low. For writes, the respective active bits in byte_lane[3:0] are low. Note: Setting this bit to 0 disables the write enable signal. WE_n will always be set to 1 (that is, you must use byte lane select signals). The byte lane state bit (PB) enables different types of memory to be connected. For byte-wide static memories, the byte_lane[3:0] signal from the memory controller is usually connected to WE_n (write enable). In this case, for reads, all byte_lane[3:0] bits must be high, which means that the byte lane state bit must be low. 16-bit wide static memory devices usually have the byte_lane[3:0] signals connected to the nUB and nLB (upper byte and lower byte) signals in the static memory. In this case, a write to a particular byte must assert the appropriate nUB or nLB signal low. For reads, all nUB and nLB signals must be asserted low so the bus is driven. In this case, the byte lane state must be high.
D06	R/W	PC	**Chip select polarity** 0 Active low chip select 1 Active high chip select
D05:04	N/A	Reserved	N/A (do not modify)
D03	R/W	PM	**Page mode** 0 Disabled (reset on reset_n) 1 Async page mode enabled (page length four) In page mode, the memory controller can burst up to four external accesses. Devices with asynchronous page mode burst four or higher are supported. Asynchronous page mode burst two devices are not supported and must be accessed normally.

Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Description
D02	R/W	BMODE	**Burst mode** Allows the static output enable signal to toggle during bursts. 0 Do not toggle output enable during bursts 1 Toggle output enable during bursts
D01:00	R/W	MW	**Memory width** 00 8 bit (reset value for chip select 0, 2, and 3 on reset_n) 01 16 bit 10 32 bit 11 Reserved The value of the chip select 1 memory width field on power-on reset (reset_n) is determined by the gpio_a[0], addr[23] signal. This value can be overridden by software. Note: For chip select 1, the value of the gpio_a[0], addr[23] signal is reflected in this field. When programmed, this register reflects the last value written into it.

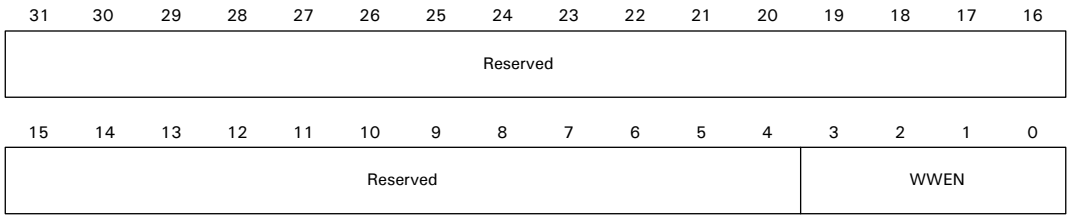
Note: Synchronous burst mode memory devices are not supported.

StaticMemory Write Enable Delay 0-3 registers

Address: A070 0204 / 0224 / 0244 / 0264

The Static Memory Write Enable Delay 0–3 registers allow you to program the delay from the chip select to the write enable assertion. The Static Memory Write Enable Delay register is used in conjunction with the Static Memory Write Delay registers, to control the width of the write enable signals. It is recommended that these registers be modified during system initialization, or when there are no current or outstanding transactions. Wait until the memory controller is idle, then enter low-power or disabled mode.

Register



Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

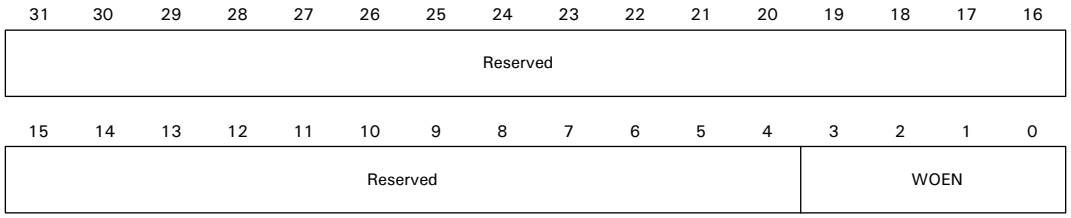
Bits	Access	Mnemonic	Description
D31:04	N/A	Reserved	N/A (do not modify)
D03:00	R/W	WWEN	**Wait write enable (WAITWEN)** 0000 One clk_out cycle delay between assertion of chip select and write enable (reset value on reset_n). 0001–1111 (n+1) clk_out cycle delay, where the delay is (WAITWEN+1) x t_{clk_out} Delay from chip select assertion to write enable.

Static Memory Output Enable Delay 0-3 registers

Address: A070 0208 / 0228 / 0248 / 0268

The Static Memory Output Enable Delay 0–3 registers allow you to program the delay from the chip select or address change, whichever is later, to the output enable assertion. The Static Memory Output Enable Delay register is used in conjunction with the Static Memory Read Delay registers, to control the width of the output enable signals. It is recommended that these registers be modified during system initialization, or when there are no current or outstanding transactions. Wait until the memory controller is idle, then enter low-power or disabled mode.

Register



Register bit assignment

Bits	Access	Mnemonic	Description
D31:04	N/A	Reserved	N/A (do not modify)
D03:00	R/W	WOEN	**Wait output enable (WAITOEN)** 0000 No delay (reset value on reset_n). 0001–1111n cycle delay, where the delay is WAITOEN x t_{clk_out} Delay from chip select assertion to output enable.

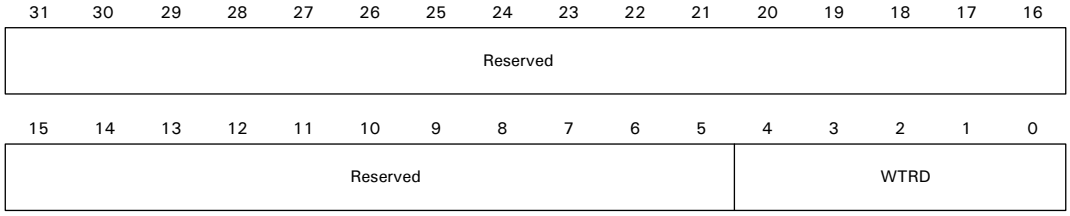
Downloaded from www.Manualslib.com manuals search engine

Static Memory Read Delay 0-3 registers

Address: A070 020C / 022C / 024C / 026C

The Static Memory Read Delay 0–3 registers allow you to program the delay from the chip select to the read access. It is recommended that these registers be modified during system initialization, or when there are no current or outstanding transactions. Wait until the memory controller is idle, then enter low-power or disabled mode. These registers are not used if the extended wait bit is set in the related Static Memory Configuration register.

Register



Register bit assignment

Bits	Access	Mnemonic	Description
D31:05	N/A	Reserved	N/A (do not modify)
D04:00	R/W	WTRD	**Nonpage mode read wait states or asynchronous page mode read first access wait state (WAITRD)** 00000–11110 (n+1) clk_out cycle for read accesses. For nonsequential reads, the wait state time is (WAITRD+1) x t_{clk_out} 11111 32 clk_out cycles for read accesses (reset value on reset_n) Use this equation to compute this field: WTRD = ([T_b + T_a + 10.0] / T_c) - 1 T_b = Total board propagation delay, including any buffers T_a = Peripheral access time T_c = clk_out clock period. Any decimal portion must be rounded up. All values are in nanoseconds

StaticMemory Page Mode Read Delay 0-3 registers

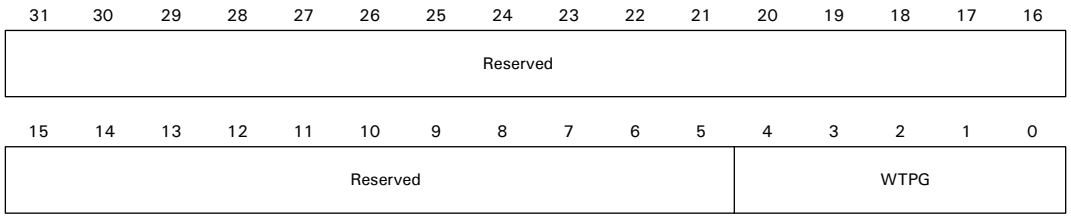
Address: A070 0210 / 0230 / 0250 / 0270

The Static Memory Page Mode Read Delay 0–3 registers allow you to program the delay for asynchronous page mode sequential accesses. These registers control the overall period for the read cycle. It is recommended that these registers be

Downloaded from www.Manualslib.com manuals search engine

modified during system initialization, or when there are no current or outstanding transactions. Wait until the memory controller is idle, then enter low-power or disabled mode.

Register



Register bit assignment

Bits	Access	Mnemonic	Description
D31:05	N/A	Reserved	N/A (do not modify)
D04:00	R/W	WTPG	**Asynchronous page mode read after the first wait state (WAITPAGE)** 00000–11110 (n+1) clk_out cycle for read access time. For asynchronous page mode read for sequential reads, the wait state time for page mode accesses after the first read is (WAITPAGE+1) x t_{clk_out} 11111 32 clk_out cycles read access time (reset value on reset_n) Number of wait states for asynchronous page mode read accesses after the first read.

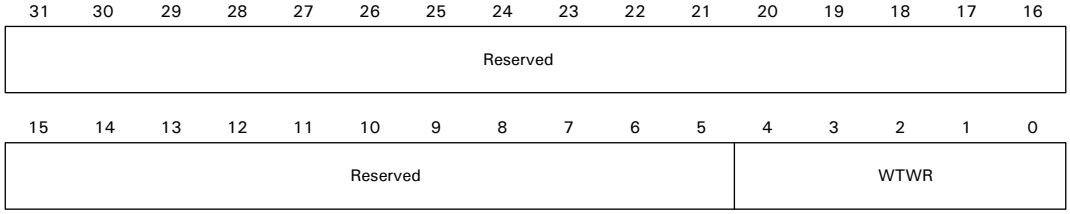
Static Memory Write Delay 0–3 registers

Address: A070 0214 / 0234 / 0254 / 0274

The Static Memory Write Delay 0–3 registers allow you to program the delay from the chip select to the write access. These registers control the overall period for the write cycle. It is recommended that these registers be modified during system initialization, or when there are no current or outstanding transactions. Wait until the memory controller is idle, then enter low-power or disabled mode.These registers are not used if the extended wait bit is enabled in the related Static Memory Configuration register.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

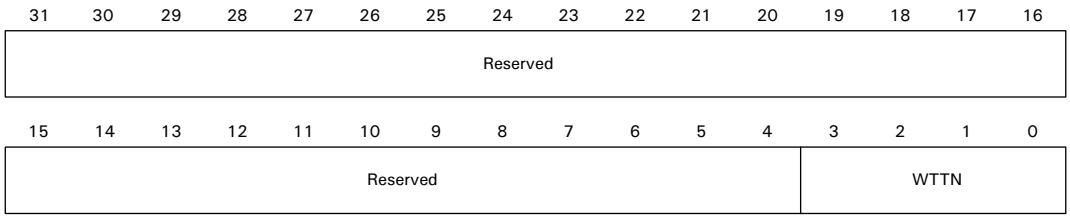
Bits	Access	Mnemonic	Description
D31:05	N/A	Reserved	N/A (do not modify)
D04:00	R/W	WTWR	**Write wait states (WAITWR)** 00000–11110 (n+2) clk_out cycle write access time. The wait state time for write accesses after the first read is WAITWR (n+2) x t_{clk_out} 11111 332 clk_out cycle write access time (reset value on reset_n) SRAM wait state time for write accesses after the first read.

StaticMemory Turn Round Delay 0–3 registers

Address: A070 0218 / 0238 / 0258 / 0278

The Static Memory Turn Round Delay 0–3 registers allow you to program the number of bus turnaround cycles. It is recommended that these registers be modified during system initialization, or when there are no current or outstanding transactions. Wait until the memory controller is idle, then enter low-power or disabled mode.

Register



Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

Bits	Access	Mnemonic	Description
D31:04	N/A	Reserved	N/A (do not modify)
D03:00	R/W	WTTN	**Bus turnaround cycles (WAITTURN)** 00000–11110 (n+1) clk_out turnaround cycles, where bus turnaround time is (WAITTURN+1) x t_{clk_out} 1111 16 clk_out turnaround cycles (reset value on reset_n).

To prevent bus contention on the external memory databus, the WAITTURN field controls the number of bus turnaround cycles added between static memory read and write accesses.

The WAITTURN field also controls the number of turnaround cycles between static memory and dynamic memory accesses.

Downloaded from www.Manualslib.com manuals search engine

Downloaded from www.Manualslib.com manuals search engine

Ethernet Communication Module

CHAPTER 6

The Ethernet Communication module consists of an Ethernet Media Access Controller (MAC) and Ethernet front-end module. The Ethernet MAC interfaces to an external PHY through the industry-standard interface: Media Independent Interface (MII). The Ethernet front-end module provides all of the control functions to the MAC.

Features

The Ethernet MAC module provides the following:

- Station address logic (SAL)
- Statistics module
- Interface to MII (Media Independent Interface) PHY

The Ethernet front-end module does the following:

- Provides control functions to the MAC
- Buffers and filters the frames received from the MAC
- Pumps transmit data into the MAC
- Moves frames between the MAC and the system memory
- Reports transmit and receive status to the host

Common acronyms

RX_RD = Receive read

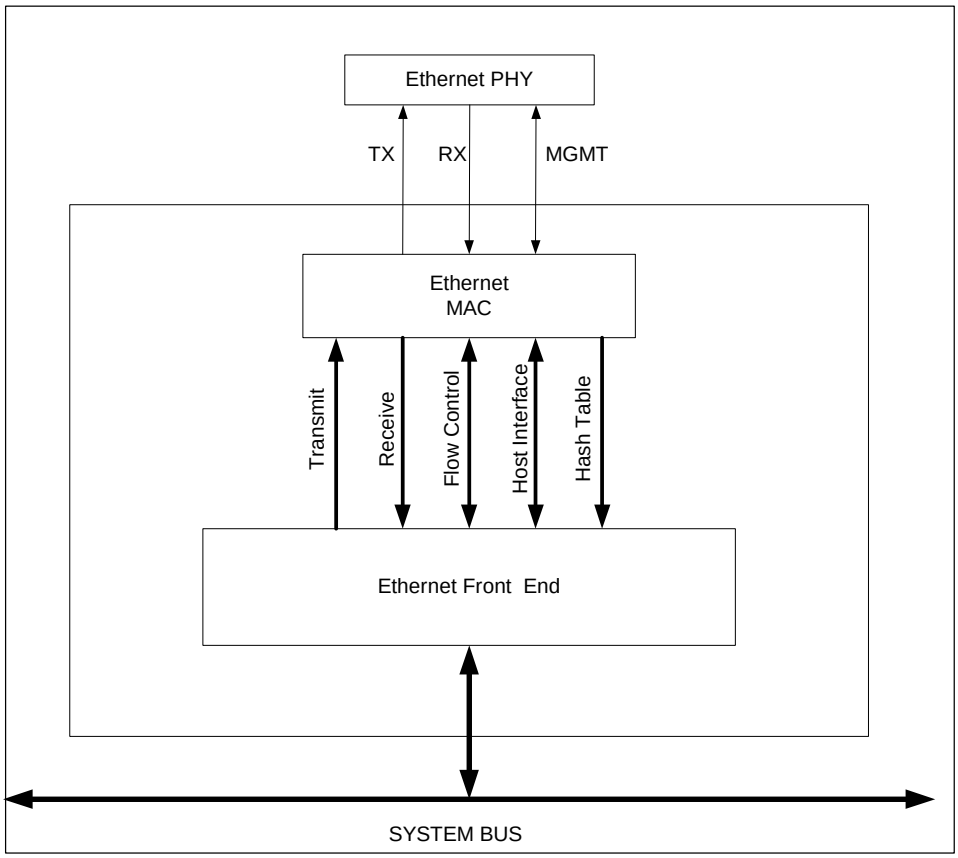
RX_WR = Receive write

TX_RD = Transmit read

TX_WR = Transmit write

Downloaded from www.Manualslib.com manuals search engine

Ethernet communications module

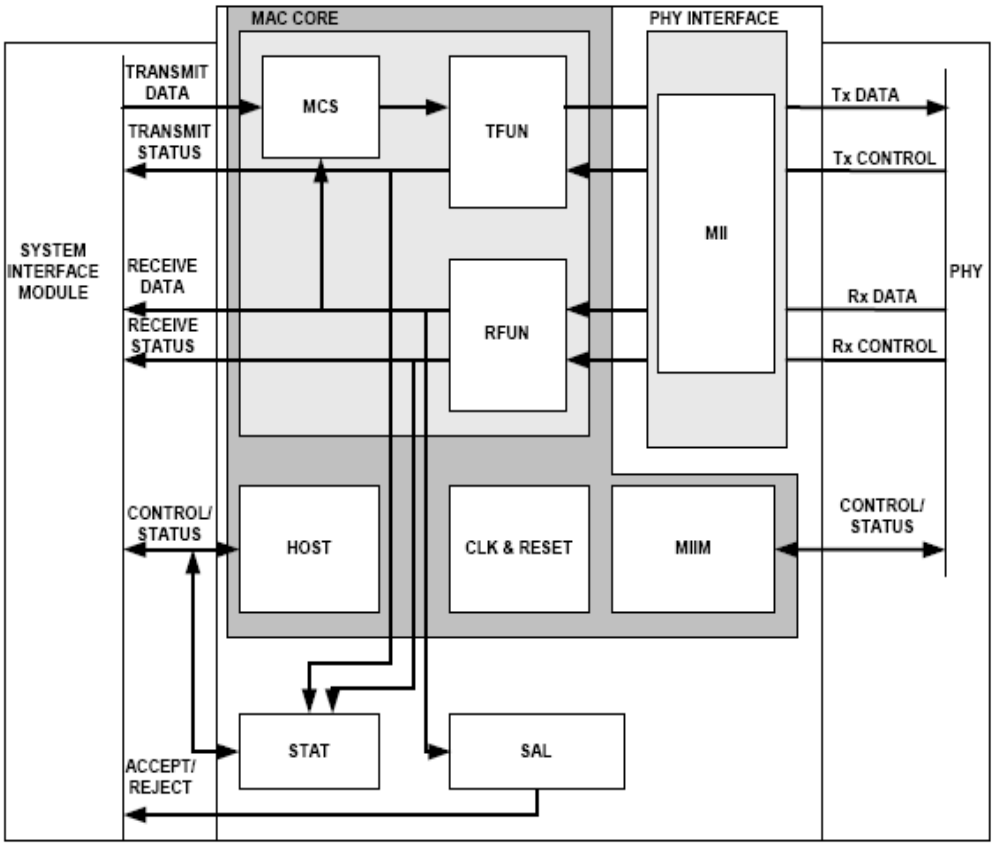


Ethernet MAC

The Ethernet MAC includes a full function 10/100 Mbps Media Access Controller (MAC), station address filtering logic (SAL), statistic collection module (STAT), and MII.

Downloaded from www.Manualslib.com manuals search engine

MAC module block diagram



MAC module features

Feature	Description
MAC Core	**10/100 megabit Media Access Controller** Performs the CSMA/CD function. ■ MCS: MAC control sublayer ■ TFUN: Transmit function ■ RFUN: Receive function
HOST	**Host interface** Provides an interface for control and configuration.
CLK & Reset	**Clocks & resets** Provides a central location for clock trees and reset logic.
MIIM	**MII management** Provides control/status path to MII PHYs.
STAT	**Statistics module** Counts and saves Ethernet statistics.

Downloaded from www.Manualslib.com manuals search engine

Feature	Description
SAL	**Station address logic** Performs destination address filtering.
MII	**Media Independent Interface** Provides the interface from the MAC core to a PHY that supports the MII (as described in the IEEE 802.3 standard).

PHY interface mappings

This table shows how the different PHY interfaces are mapped to the external IO.

External IO	MII
RXD[3]	RXD[3]
RXD[2]	RXD[2]
RXD[1]	RXD[1]
RXD[0]	RXD[0]
RX_DV	RX_DV
RX_ER	RX_ER
RX_CLK	RX_CLK
TXD[3]	TXD[3]
TXD[2]	TXD[2]
TXD[1]	TXD[1]
TXD[0]	TXD[0]
TX_EN	TX_EN
TX_ER	TX_ER
TX_CLK	TX_CLK
CRS	CRS
COL	COL
MDC	MDC
MDIO	MDIO

Station address logic (SAL)

The station address logic module examines the destination address field of incoming frames, and filters the frames before they are stored in the Ethernet front-end

Downloaded from www.Manualslib.com manuals search engine

module. The filtering options, listed next, are programmed in the Station Address Filter register (see page 301).

- Accept frames to destination address programmed in the SA1, SA2, and SA3 registers (Station Address registers, beginning on page 300)
- Accept all frames
- Accept all multicast frames
- Accept all multicast frames using HT1 and HT2 registers. See “Sample hash table code,” on page 334)
- Accept all broadcast frames

The filtering conditions are independent of each other; for example, the Station Address Logic register can be configured to accept all broadcast frames, and frames to the programmed destination address.

MAC receiver

- The MAC receiver provides the station address logic with a 6-bit CRC value that is the upper 6 bits of a 32-bit CRC calculation performed on the 48-bit multicast destination address. This 6-bit value addresses the 64-bit multicast hash table created in the HT1 and HT2 registers. See “Sample hash table code,” on page 334)
- If the current receive frame is a multicast frame and the 6-bit CRC addresses a bit in the hash table that is set to 1, the receive frame is accepted; otherwise, the frame is rejected. See “Sample hash table code,” on page 334) for sample C code to calculate hash table entries.

Statistics module

The Statistics module counts and saves Ethernet statistics in several counters (see “Statistics registers” on page 303).

The Ethernet General Control Register #2 contains three statistics module configuration bits:

- **AUTOZ.** Enable statistics counter clear on read.
- **CLRCNT.** Clear statistics counters.
- **STEN.** Enable statistics counters.

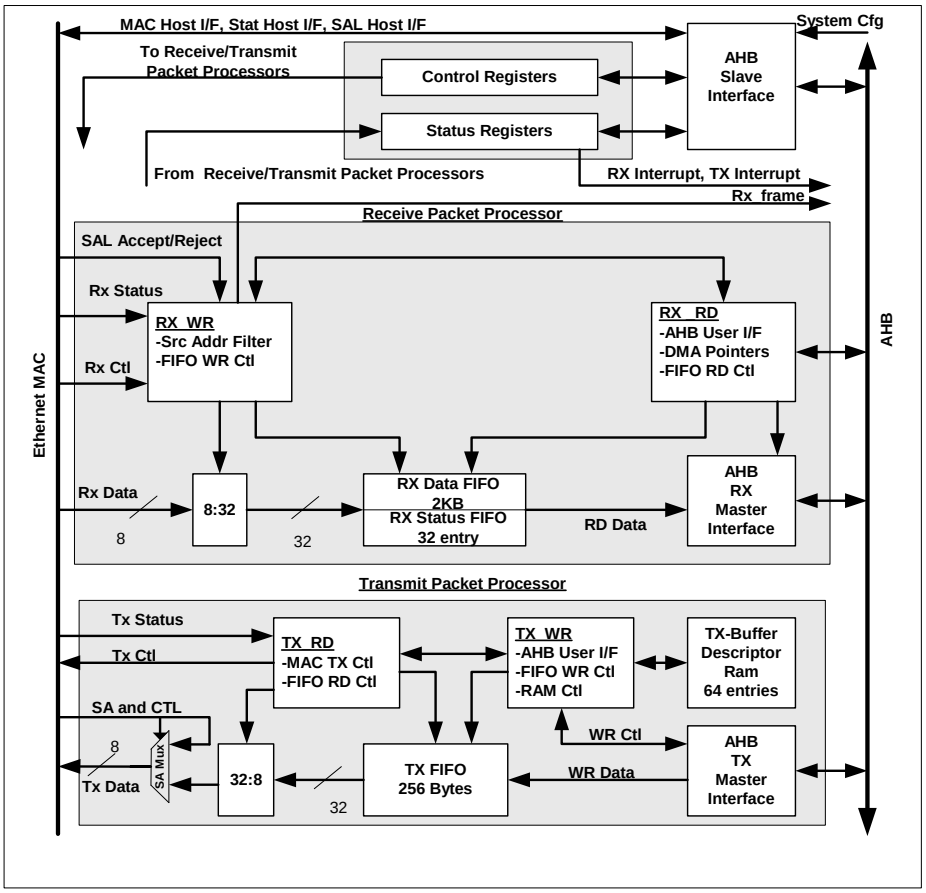
If any of the counters roll over, an associated carry bit is set in the Carry 1 (CAR1) or Carry 2 (CAR2) registers (see "General Statistics registers address map," beginning on page 310). Any statistics counter overflow can cause the STOVFL bit in the Ethernet Interrupt Status register (see page 317) to be set if its associated mask bit is not set in Carry Mask Register 1 or Carry Mask Register 2.

Downloaded from www.Manualslib.com manuals search engine

The counters support a *clear on read* capability that is enabled when AUTOZ is set to 1 in the Ethernet General Control Register #2.

Ethernet front-end module

Ethernet front-end module (EFE)



The EFE module includes a set of control and status registers, a receive packet processor, and a transmit packet processor. On one side, the Ethernet front end interfaces to the MAC and provides all control and status signals required by the MAC. On the other side, the Ethernet front end interfaces to the system.

Receive packet processor

The receive packet processor accepts good Ethernet frames (for example, valid checksum and size) from the Ethernet MAC and commits them to external system memory. Bad frames (for example, invalid checksum or code violation) and frames with unacceptable destination addresses are discarded.

Downloaded from www.Manualslib.com manuals search engine

The 2K byte RX_FIFO allows the entire Ethernet frame to be buffered while the receive byte count is analyzed. The receive byte count is analyzed by the receive packet processor to select the optimum-sized buffer for transferring the received frame to system memory. The processor can use one of four different-sized receive buffers in system memory.

Transmit packet processor

The transmit packet processor transfers frames constructed in system memory to the Ethernet MAC. The software initializes a buffer descriptor table in a local RAM that points the transmit packet processor to the various frame segments in system memory. The 256-byte TX_FIFO decouples the data transfer to the Ethernet MAC from the AHB bus fill rate.

Receive packet processor

As a frame is received from the Ethernet MAC, it is stored in the receive data FIFO. At the end of the frame, an accept/reject decision is made based on several conditions. If the packet is rejected, it is flushed from the receive data FIFO.

If a frame is accepted, status signals from the MAC, including the receive size of the frame, are stored in a separate 32-entry receive status FIFO; the RX_RD logic is notified that a good frame is in the FIFO.

If the RX_WR logic tries to write to a full receive data FIFO anytime during the frame, it flushes the frame from the receive data FIFO and sets RXOVFL_DATA (RX data FIFO overflowed) in the Ethernet Interrupt Status register. For proper operation, reset the receive packet processor using the ERX bit in the Ethernet General Control Register #1 when this condition occurs. If the RX_WR logic tries to write a full receive status FIFO at the end of the frame, the RX_WR logic flushes the frame from the receive data FIFO and sets RXOVFL_STAT (RX status FIFO overflowed) in the Ethernet Interrupt Status register.

Power down mode

The RX_WR logic supports the processor system power down and recovery functionality. In this mode, the RX clock to the MAC and the RX_WR logic are still active, but the clock to the RX_RD and AHB interface is disabled. This allows frames to be received and written into the receive FIFO, but the frame remains in the FIFO until the system wakes up. Normal frame filtering is still performed.

When a qualified frame is inserted into the receive FIFO, the receive packet processor notifies the system power controller, which performs the *wake up* sequence. The frame remains in the receive FIFO until the system wakes up.

Downloaded from www.Manualslib.com manuals search engine

Transferring a frame to system memory

The RX_RD logic manages the transfer of a frame in the RX_FIFO to system memory. The transfer is enabled by setting the ERXDMA (enable receive DMA) bit in Ethernet General Control Register #1.

Transferring a frame in the receive FIFO to system memory begins when the RX_WR logic notifies the RX_RD logic that a good frame is in the receive FIFO. Frames are transferred to system memory using up to four rings (that is, 1, 2, or 3 rings can also be used) of buffer descriptors that point to buffers in system memory. The maximum frame size that each ring can accept is programmable. The first thing the RX_RD logic does, then, is analyze the frame length in the receive status FIFO to determine which buffer descriptor to use.

The RX_RD logic goes through the four buffer descriptors looking for the optimum buffer size. It searches the enabled descriptors starting with A, then B, C, and finally D; any pools that are full (that is, the F bit is set in the buffer descriptor) are skipped. The search stops as soon as the logic encounters an available buffer that is large enough to hold the entire receive frame.

The pointers to the first buffer descriptor in each of the four pools are found in the related Buffer Descriptor Pointer register (RXAPTR, RXBPTR, RXCPTR, RXDPTR). Pointers to subsequent buffer descriptors are generated by adding an offset of 0x10 from this pointer for each additional buffer used.

Receive buffer descriptor format

	31	30	29	28		16	15		0
OFFSET + 0	Source Address								
OFFSET + 4	Buffer Length (11 lower bits used)								
OFFSET + 8	Destination Address (not used)								
OFFSET + C	W	I	E	F	Reserved		Status		

Receive buffer descriptor format description

The current buffer descriptor for each pool is kept in local registers. The current buffer descriptor registers are initialized to the buffer descriptors pointed to by the Buffer Descriptor Pointer registers, by setting the ERXINIT (enable initialization of RX buffer descriptor registers) bit in Ethernet General Control Register #1. The initialization process is complete when RXINIT (RX initialization complete) is set in the Ethernet General Status register. At the end of a frame, the next buffer descriptor for the ring just used is read from system memory and stored in the registers internal to the RX_RD logic.

Downloaded from www.Manualslib.com manuals search engine

Receive buffer descriptor field definitions

Field	Description
W	WRAP bit, which, when set, tells the RX_RD logic that this is the last buffer descriptor in the ring. In this situation, the next buffer descriptor is found using the appropriate Buffer Descriptor Pointer register. When the WRAP bit is not set, the next buffer descriptor is found using an offset of 0x10 from the current buffer descriptor pointer.
I	When set, tells the RX_RD logic to set RXBUFC in the Ethernet Interrupt Status register after the frame has been transferred to system memory.
E	ENABLE bit, which, when set, tells the RX_RD logic that this buffer descriptor is enabled. When a new frame is received, pools that do not have the ENABLE bit set in their next buffer descriptor are skipped when deciding in which pool to put the frame. The receive processor can use up to four different-sized receive buffers in system memory. **Note:** To enable a pool that is currently disabled, change the ENABLE bit from 0 to 1 and reinitialize the buffer descriptors pointed to by the Buffer Descriptor Pointer register: 1 Set the ERXINIT bit in the Ethernet General Control Register 1. 7 Wait for RXINIT to be set in the Ethernet General Status register. Change the ENABLE bit only while the receive packet processor is idle.
Buffer pointer	32-bit pointer to the start of the buffer in system memory. This pointer must be aligned on a 32-bit boundary.
Status	Lower 16 bits of the Ethernet Receive Status register. The status is taken from the receive status FIFO and added to the buffer descriptor after the last word of the frame is written to system memory.
F	When set, indicates the buffer is full. The RX_RD logic sets this bit after filling a buffer. The system software clears this bit, as required, to free the buffer for future use. When a new frame is received, pools that have the F bit set in their next buffer descriptor are skipped when deciding in which pool to put the frame.
Buffer length	This is a dual use field: ■ When the buffer descriptor is read from system memory, buffer length indicates the maximum sized frame, in bytes, that can be stored in this buffer ring. ■ When the RX_RD logic writes the descriptor back from the receive status FIFO into system memory at the end of the frame, the buffer length is the actual frame length, in bytes.Only the lower 11 bits of this field are valid, since the maximum legal frame size for Ethernet is 1522 bytes.

Transmit packet processor

Transmit frames are transferred from system memory to the transmit packet processor into a 256-byte TX_FIFO. Because various parts of the transmit frame can

Downloaded from www.Manualslib.com manuals search engine

reside in different buffers in system memory, several buffer descriptors can be used to transfer the frame.

Transmit buffer descriptor format

All buffer descriptors (that is, up to 64) are found in a local TX buffer descriptor RAM. This is the transmit buffer descriptor format.

	31	30	29	28		16	15		0
OFFSET + 0	Source Address								
OFFSET + 4	Buffer Length (11-bits used)								
OFFSET + 8	Destination Address (not used)								
OFFSET + C	W	I	L	F	Reserved		Status		

Transmit buffer descriptor field definitions

Field	Description
W	WRAP bit, which, when set, tells the TX_WR logic that this is the last buffer descriptor within the continuous list of descriptors in the TX buffer descriptor RAM. The next buffer descriptor is found using the initial buffer descriptor pointer in the TX Buffer Descriptor Pointer register (TXPTR). When the WRAP bit is not set, the next buffer descriptor is located at the next entry in the TX buffer descriptor RAM.
I	When set, tells the TX_WR logic to set TXBUFC in the Ethernet Interrupt Status register when the buffer is closed due to a normal channel completion.
Buffer pointer	32-bit pointer to the start of the buffer in system memory. This pointer can be aligned on any byte of a 32-bit word.
Status	Lower 16 bits of the Ethernet Transmit Status register. The status is returned from the Ethernet MAC at the end of the frame and written into the last buffer descriptor of the frame.
L	When set, tells the TX_WR logic that this buffer descriptor is the last descriptor that completes an entire frame. This bit allows multiple descriptors to be chained together to make up a frame.

Downloaded from www.Manualslib.com manuals search engine

Field	Description
F	When set, indicates the buffer is full. The TX_WR logic clears this bit after emptying a buffer. The system software sets this bit as required, to signal that the buffer is ready for transmission. If the TX_WR logic detects that this bit is not set when the buffer descriptor is read, it does one of two things: ■ If a frame is not in progress, the TX_WR logic sets the TXIDLE bit in the Ethernet Interrupt Status register. ■ If a frame is in progress, the TXBUFNR bit in the Ethernet Interrupt Status register is set. In either case, the TX_WR logic stops processing frames until TCLER (clear transmit logic) in Ethernet General Control Register #2 is toggled from low to high. TXBUFNR is set only for frames that consist of multiple buffer descriptors and contain a descriptor — *not* the first descriptor — that does not have the F bit set after frame transmission has begun.
Buffer length	This is a dual use field: ■ When the buffer descriptor is read from the TX buffer descriptor RAM, buffer length indicates the length of the buffer, in bytes. The TX_WR logic uses this information to identify the end of the buffer. For proper operation of the TX_WR logic, all transmit frames must be at least 34 bytes in length. ■ When the TX_WR logic updates the buffer descriptor at the end of the frame, it writes the length of the frame, in bytes, into this field for the last buffer descriptor of the frame. If the MAC is configured to add the CRC to the frame (that is, CRCEN in MAC Configuration Register #2 is set to 1), this field will include the four bytes of CRC. This field is set to 0x000 for jumbo frames that are aborted. Only the lower 11 bits of this field are valid, since the maximum legal frame size for Ethernet is 1522 bytes.

Transmitting a frame

Setting the EXTDMA (enable transmit DMA) bit in Ethernet General Control Register #1 starts the transfer of transmit frames from the system memory to the TX_FIFO. The TX_WR logic reads the first buffer descriptor in the TX buffer descriptor RAM.

- If the F bit is set, it transfers data from system memory to the TX_FIFO using the buffer pointer as the starting point. This process continues until the end of the buffer is reached. The address for each subsequent read of the buffer is incremented by 32 bytes (that is, 0x20). The buffer length field in the buffer descriptor is decremented by this same value, each transfer, to identify when the end of the buffer is reached.
- If the L field in the buffer descriptor is 0, the next buffer descriptor in the RAM continues the frame transfer until the L field in the current buffer descriptor is 1. This identifies the current buffer as the last buffer of a transmit frame.

After the entire frame has been written to the TX_FIFO, the TX_WR logic waits for a signal from the TX_RD logic indicating that frame transmission has completed at the MAC. The TX_WR logic updates the buffer length, status, and F fields of the current buffer descriptor (that is, the last buffer descriptor for the frame) in the TX buffer descriptor RAM when the signal is received.

Downloaded from www.Manualslib.com manuals search engine

The TX_WR logic examines the status received from the MAC after it has transmitted the frame.

Frame transmitted successfully

If the frame was transmitted successfully, the TX_WR logic sets TXDONE (frame transmission complete) in the Ethernet Interrupt Status register and reads the next buffer descriptor. If a new frame is available (that is, the F bit is set), the TX_WR starts transferring the frame. If a new frame is not available, the TX_WR logic sets the TXIDLE (TX_WR logic has no frame to transmit) bit in the Ethernet Interrupt Status register and waits for the software to toggle TCLER (clear transmit logic), in Ethernet General Control Register #2, from low to high to resume processing. When TCLER is toggled, transmission starts again with the buffer descriptor pointed to by the Transmit Recover Buffer Descriptor Pointer register. Software should update this register before toggling TCLER.

Frame transmitted unsuccessfully

If the TX_WR logic detects that the frame was aborted or had an error, the logic updates the current buffer descriptor as described in the previous paragraph. If the frame was aborted before the last buffer descriptor of the frame was accessed, the result is a situation in which the status field of a buffer descriptor, which is not the last buffer descriptor in a frame, has a non-zero value. The TX_WR logic stops processing frames until TCLER (clear transmit logic) in Ethernet General Control Register #2 is toggled from low to high to resume processing. The TX_WR logic also sets TXERR (last frame not transmitted successfully) in the Ethernet Interrupt Status register and loads the TX buffer descriptor RAM address of the current buffer descriptor in the TX Error Buffer Descriptor Pointer register (see page 320). This allows identification of the frame that was not transmitted successfully. As part of the recovery procedure, software must read the TX Error Buffer Descriptor Pointer register and then write the 8-bit address of the buffer descriptor to resume transmission into the TX Recover Buffer Descriptor Pointer register.

Transmitting a frame to the Ethernet MAC

The TX_RD logic is responsible for reading data from the TX_FIFO and sending it to the Ethernet MAC. The logic does not begin reading a new frame until the TX_FIFO is full. This scheme decouples the data transfer to the Ethernet MAC from the fill rate from the AHB bus. For short frames that are less than 256 bytes, the transmit process begins when the end-of-frame signal is received from the TX_WR logic.

When the MAC completes a frame transmission, it returns status bits that are stored in the Ethernet Transmit Status register (see page 283) and written into the status field of the current buffer descriptor.

Ethernet underrun

An Ethernet underrun can only occur due to the following programming errors:

- Insufficient bandwidth is assigned to the Ethernet transmitter.

Downloaded from www.Manualslib.com manuals search engine

– A packet consisting of multiple, linked buffer descriptors does not have the F bit set in any of the non-first buffer descriptors.

When an underrun occurs, it is also possible for the Ethernet transmitter to send out a corrupted packet with a good Ethernet CRC if the MAC is configured to add the CRC to the frame (that is, CRCEN in MAC Configuration Register #2 is set to 1).

Ethernet slave interface

The AHB slave interface supports only single 32-bit transfers. The slave interface also supports limiting CSR and RAM accesses to CPU "privileged mode" accesses. Use the internal register access mode bit 0 in the Miscellaneous System Configuration register to set access accordingly (see "Miscellaneous System Configuration and Status register," beginning on page 184).

The slave also generates an AHB ERROR if the address is not aligned on a 32-bit boundary, and the misaligned bus address response mode is set in the Miscellaneous System Configuration register. In addition, accesses to non-existent addresses result in an AHB ERROR response.

Interrupts

Separate RX and TX interrupts are provided back to the system.

Interrupt sources

This table shows all interrupt sources and the interrupts to which they are assigned.

Interrupt condition	Description	Interrupt
RX data FIFO overflow	RX data FIFO overflowed. For proper operation, reset the receive packet processor using the ERX bit in the Ethernet General Control Register #1 when this condition occurs.	RX
RX status FIFO overflow	RX status overflowed.	RX
Receive buffer closed	I bit set in receive buffer descriptor and buffer closed.	RX
Receive complete (Pool A)	Complete receive frame stored in pool A of system memory.	RX
Receive complete (Pool B)	Complete receive frame stored in pool B of system memory.	RX
Receive complete (Pool C)	Complete receive frame stored in pool C of system memory.	RX
Receive complete (Pool D)	Complete receive frame stored in pool D of system memory.	RX

Downloaded from www.Manualslib.com manuals search engine

Interrupt condition	Description	Interrupt
No receive buffers	No buffer is available for this frame because all 4 buffer rings are disabled, full, or no available buffer is big enough for the frame.	RX
Receive buffers full	No buffer is available for this frame because all 4 buffers are disabled or full.	RX
RX buffer ready	Frame available in RX_FIFO. (Used for diagnostics.)	RX
Statistics counter overflow	One of the statistics counters has overflowed. Individual counters can be masked using the CAM1 and CAM2 registers.	TX
Transmit buffer closed	I bit set in Transmit buffer descriptor and buffer closed.	TX
Transmit buffer not ready	F bit not set in transmit buffer descriptor when read from TX buffer descriptor RAM, for a frame in progress.	TX
Transmit complete	Frame transmission complete.	TX
TXERR	Frame not transmitted successfully.	TX
TXIDLE	TX_WR logic in idle mode because there are no frames to send.	TX

Status bits

The status bits for all interrupts are available in the Ethernet Interrupt Status register, and the associated enables are available in the Ethernet Interrupt Enable register. Each interrupt status bit is cleared by writing a 1 to it.

Resets

This table provides a summary of all resets used for the Ethernet front-end and MAC, as well as the modules the resets control.

Bit field	Register	Active state	Default state	Modules reset
ERX	Ethernet General Control Register #1	0	0	RX_RD, RX_WR
ETX	Ethernet General Control Register #1	0	0	TX_RD, TX_WR
MAC_HRST	Ethernet General Control Register #1	1	0	MAC, STAT, RX_WR, TX_RD, programmable registers in Station Address Logic
SRST	MAC1	1	1	MAC (except programmable registers), Station Address Logic (except programmable registers), RX_WR, TX_RD
RPERFUN	MAC1	1	0	MAC RX logic
RPEMCST	MAC1	1	0	MAC PEMCS (TX side)

Downloaded from www.Manualslib.com manuals search engine

Bit field	Register	Active state	Default state	Modules reset
RPETFUN	MAC1	1	0	MAC TX logic
MIIM	MII Management Configuration register	1	0	MAC MIIM logic

Multicast address filtering

The RX-WR logic contains a programmable 8-entry multicast address filter that provides more restrictive filtering than that available in the MAC using the SAL. Only multicast addresses that match those programmed into the filter will be accepted.

Filter entries

Each entry in the filter consists of a 48-bit destination address, an enable bit, and a 48-bit mask. The mask contains a 1 in each bit position of the address that is used in the address filter.; this is used to extend the range of each entry.

Multicast address filter registers

Register	Description
MFILTL [7:0]	Lower 32 bits of multicast address
MFILTH [7:0]	Upper 16 bits of multicast address
MCMSKL	Lower 32 bits of multicast address mask
MCMSKH [7:0]	Upper 16 bits of multicast address
MFILTEN	Per-entry enable bits

Multicast address filtering example 1

To accept only multicast packets with destination address 0x01_00_5E_00_00_00 using entry 0, the registers are set as shown:

Register	Value	Function
MFILTEN	0x1	Enable entry 0
MFILTL0	0x5E_00_00_00	Lower 32 bits of multicast address
MFILTH0	0x01_00	Upper 16 bits of multicast address
MCMSKL0	0xFFFF_FFFF	Include all of the lower 32 bits of the multicast address in the comparison.
MCMSKH0	0xFFFF	Include all of the upper 16 bits of the multicast address in the comparison

Downloaded from www.Manualslib.com manuals search engine

Multicast address filtering example 2

To accept multicast packets with destination addresses in the range of 0x01_00_5E_00_00_00 to 0x01_00_5E_00_00_0f using entry 4, the registers are set as shown:

Register	Value	Function
MFILTEN	0x10	Enable entry 4
MFILTL4	0x5E_00_00_00	Lower 32 bits of multicast address
MFILTH4	0x01_00	Upper 16 bits of multicast address
MCMSKL4	0xFFFF_FFF0	Include only bits [31:04] of the lower 32 bits of the multicast address in the comparison.
MCMSKH4	0xFFFF	Include all of the upper 16 bits of the multicast address in the comparison

Notes

- If any of the address filter entries are enabled, the SAL must be set up to accept all multicast packets by setting the PRM bit in the Station Address Filter register.
- Runt packets that are less than 6 bytes, and therefore do not have a valid destination address, are automatically discarded by the multicast address filtering logic.

Clock synchronization

The multicast filtering logic resides in the RX CLK domain, but all of the registers are controlled in the AHB clock domain. To provide traditional dual-rank clock synchronization flops for each bit of the five Multicast Address Filter registers consumes a large amount of gates. Therefore, the logic is designed such that only the MFILTEN register bits are synchronized and when these bits are cleared, changes in the other register values are not seen at the input of any internal flops in the RX CLK domain.

Writing to other registers

Use these steps to dynamically write to any of the other Multicast Address Filter registers:

1. Clear the enable bit in the MFILTEN register for the address filter you want to change.
2. Update the address filter registers for the disable filter.
3. Set the enable bit for the address filter that was just changed.

If the address filters are changed only when the RX_WR logic is reset or not processing frames, as recommended, the address filter registers can be updated without using this procedure.

Downloaded from www.Manualslib.com manuals search engine

Ethernet Control and Status registers

All configuration registers must be accessed as 32-bit words and as single accesses only. Bursting is not allowed.

Register address filter

Address	Register	Description
A060 0000	EGCR1	Ethernet General Control Register #1
A060 0004	EGCR2	Ethernet General Control Register #2
A060 0008	EGSR	Ethernet General Status register
A060 000C–A060 0014		Reserved
A060 0018	ETSR	Ethernet Transmit Status register
A060 001C	ERSR	Ethernet Receive Status register
A060 0400	MAC1	MAC Configuration Register #1
A060 0404	MAC2	MAC Configuration Register #2
A060 0408	IPGT	Back-to-Back Inter-Packet-Gap register
A060 040C	IPGR	Non-Back-to-Back Inter-Packet-Gap register
A060 0410	CLRT	Collision Window/Retry register
A060 0414	MAXF	Maximum Frame register
A060 0418–A060 041C		Reserved
A060 0420	MCFG	MII Management Configuration register
A060 0424	MCMD	MII Management Command register
A060 0428	MADR	MII Management Address register
A060 042C	MWTD	MII Management Write Data register
A060 0430	MRDD	MII Management Read Data register
A060 0434	MIND	MII Management Indicators register
A060 0440	SA1	Station Address Register #1
A060 0444	SA2	Station Address Register #2
A060 0448	SA3	Station Address register #3
A060 0500	SAFR	Station Address Filter register
A060 0504	HT1	Hash Table Register #1
A060 0508	HT2	Hash Table Register #2
A060 0680	STAT	Statistics Register Base (45 registers)
A060 0A00	RXAPTR	RX_A Buffer Descriptor Pointer register
A060 0A04	RXBPTR	RX_B Buffer Descriptor Pointer register
A060 0A08	RXCPTR	RX_C Buffer Descriptor Pointer register

Downloaded from www.Manualslib.com manuals search engine

Address	Register	Description
A060 0A0C	RXDPTR	RX_D Buffer Descriptor Pointer register
A060 0A10	EINTR	Ethernet Interrupt Status register
A060 0A14	EINTREN	Ethernet Interrupt Enable register
A060 0A18	TXPTR	TX Buffer Descriptor Pointer register
A060 0A1C	TXRPTR	TX Recover Buffer Descriptor Pointer register
A060 0A20	TXERBD	TX Error Buffer Descriptor Pointer register
A060 0A24	TXSPTR	TX Stall Buffer Descriptor Pointer register
A060 0A28	RXAOFF	RX_A Buffer Descriptor Pointer Offset register
A060 0A2C	RXBOFF	RX_B Buffer Descriptor Pointer Offset register
A060 0A30	RXCOFF	RX_C Buffer Descriptor Pointer Offset register
A060 0A34	RXDOFF	RX_D Buffer Descriptor Pointer Offset register
A060 0A38	TXOFF	Transmit Buffer Descriptor Pointer Offset register
A060 0A3C	RXFREE	RX Free Buffer register
A060 0A40	MFILTL0	Multicast Low Address Filter Register 0
A060 0A44	MFILTL1	Multicast Low Address Filter Register 1
A060 0A48	MFILTL2	Multicast Low Address Filter Register 2
A060 0A4C	MFILTL3	Multicast Low Address Filter Register 3
A060 0A50	MFILTL4	Multicast Low Address Filter Register 4
A060 0A54	MFILTL5	Multicast Low Address Filter Register 5
A060 0A58	MFILTL6	Multicast Low Address Filter Register 6
A060 0A5C	MFILTL7	Multicast Low Address Filter Register 7
A060 0A60	MFILTH0	Multicast High Address Filter Register 0
A060 0A64	MFILTH1	Multicast High Address Filter Register 1
A060 0A68	MFILTH2	Multicast High Address Filter Register 2
A060 0A6C	MFILTH3	Multicast High Address Filter Register 3
A060 0A70	MFILTH4	Multicast High Address Filter Register 4
A060 0A74	MFILTH5	Multicast High Address Filter Register 5
A060 0A78	MFILTH6	Multicast High Address Filter Register 6
A060 0A7C	MFILTH7	Multicast High Address Filter Register 7
A060 0A80	MFMSKL0	Multicast Low Address Mask Register 0
A060 0A84	MFMSKL1	Multicast Low Address Mask Register 1
A060 0A88	MFMSKL2	Multicast Low Address Mask Register 2
A060 0A8C	MFMSKL3	Multicast Low Address Mask Register 3

Downloaded from www.Manualslib.com manuals search engine

Address	Register	Description
A060 0A90	MFMSKL4	Multicast Low Address Mask Register 4
A060 0A94	MFMSKL5	Multicast Low Address Mask Register 5
A060 0A98	MFMSKL6	Multicast Low Address Mask Register 6
A060 0A9C	MFMSKL7	Multicast Low Address Mask Register 7
A060 0AA0	MFMSKH0	Multicast High Address Mask Register 0
A060 0AA4	MFMSKH1	Multicast High Address Mask Register 1
A060 0AA8	MFMSKH2	Multicast High Address Mask Register 2
A060 0AAC	MFMSKH3	Multicast High Address Mask Register 3
A060 0AB0	MFMSKH4	Multicast High Address Mask Register 4
A060 0AB4	MFMSKH5	Multicast High Address Mask Register 5
A060 0AB8	MFMSKH6	Multicast High Address Mask Register 6
A060 0ABC	MFMSKH7	Multicast High Address Mask Register 7
A060 0AC0	MFILTEN	Multicast Address Filter Enable Register
A060 1000	TXBD	TX Buffer Descriptor RAM (256 locations)
A060 2000	RXRAM	RX FIFO RAM (512 locations)

Ethernet General Control Register #1

Address: A060 0000

Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERX	ERXDMA	Reserved	ERXSHT	Not used				ETX	ETXDMA	Not used	Not used	ERXINIT	Reserved		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			Not used	RXSHFT	RXALIGN	MAC_HRST	ITXA	RXRAMI	Reserved						

Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31	R/W	ERX	0	**Enable RX packet processing** 0 Reset RX 1 Enable RX Used as a soft reset for the RX. When cleared, resets all logic in the RX and flushes the FIFO. The ERX bit must be set active high to allow data to be received from the MAC receiver.
D30	R/W	ERXDMA	0	**Enable receive DMA** 0 Disable receive DMA data request (use to stall receiver) 1 Enable receive DMA data request Must be set active high to allow the RX_RD logic to request the AHB bus to DMA receive frames into system memory. Set this bit to zero to temporarily stall the receive side Ethernet DMA. The RX_RD logic stalls on frame boundaries.
D29	N/A	Reserved	N/A	N/A
D28	R/W	ERXSHT	0	**Accept short (<64) receive frames** 0 Do not accept short frames 1 Accept short frames When set, allows frames that are smaller than 64 bytes to be accepted by the RX_WR logic. ERXSHT is typically set for debugging only.
D27:24	R/W	Not used	0	Always write as 0.
D23	R/W	ETX	0	**Enable TX packet processing** 0 Reset TX 1 Enable TX Used as a soft reset for the TX. When cleared resets all logic in the TX and flushes the FIFOs. ETX must be set active high to allow data to be sent to the MAC and to allow processor access to the TX buffer descriptor RAM.

Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Reset	Description
D22	R/W	ETXDMA	0	**Enable transmit DMA** 0 Disable transmit DMA data request (use to stall transmitter) 1 Enable transmit DMA data request Must be set active high to allow the transmit packet processor to issue transmit data requests to the AHB interface. Set this bit to 0 to temporarily stall frame transmission, which always stalls at the completion of the current frame. The 8-bit address of the next buffer descriptor to be read in the TX buffer descriptor RAM is loaded into the TXSPTR register when the transmit process ends. If the transmit packet processor already is stalled and waiting for TCLER, clearing ETXDMA will not take effect until TCLER has been toggled. This bit generally should be set after the Ethernet transmit parameters (for example, buffer pointer descriptor) are programmed into the transmit packet processor.
D21	R/W	Not used	1	Always write as 1.
D20	R/W	Not used	0	Always write as 0.
D19	R/W	ERXINIT	0	**Enable initialization of RX buffer descriptors** 0 Do not initialize 1 Initialize When set, causes the RX_RD logic to initialize the internal buffer descriptor registers for each of the four pools from the buffer descriptors pointed to by RXAPTR, RXBPTR, RXCPTR, and RXDPTR. This is done as part of the RX initialization process. RXINIT is set in the Ethernet General Status register when the initialization process is complete, and ERXINIT must be cleared before enabling frame reception from the MAC. The delay from ERXINIT set to RXINIT set is less than five microseconds.
D18:13	N/A	Reserved	N/A	N/A
D12	R/W	Not used	0	Always write as 0.
D11	R/W	RXSHFT	0	Shift RX data 0 Standard receive format. No padding bytes added before receive frame data 1 The receiver inserts 2 bytes of padding before the first byte of the receive data, to create longword alignment of the payload.

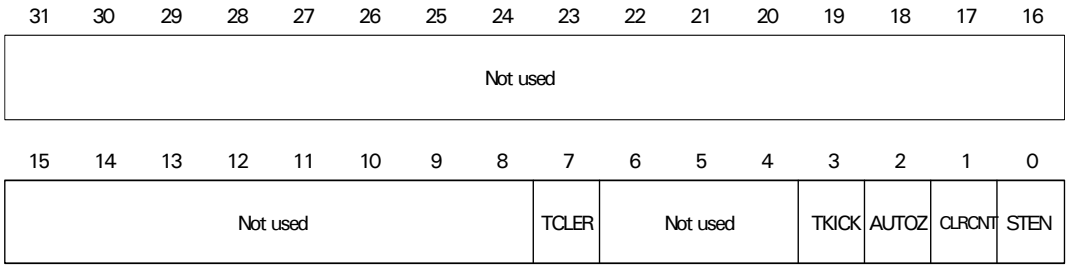
Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Reset	Description
D10	R/W	RXALIGN	0	**Align RX data** 0 Standard receive format. The data block immediately follows the 14-byte header block. 1 The receiver inserts a 2-byte padding between the 14-byte header and the data block, causing longword alignment for both the header and data blocks.
D09	R/W	MAC_HRST	1	**MAC host interface soft reset** 0 Restore MAC, STAT, SAL, RX_WR, and TX_RD to normal operation. 1 Reset MAC, STAT, programmable registers in SAL, RX_WR, and TX_RD. Keep high for minimum of 5µsec to guarantee that all functions get reset.
D08	R/W	ITXA	0	**Insert transmit source address** 0 Source address for Ethernet transmit frame taken from data in TX_FIFO. 1 Insert the MAC Ethernet source address into the Ethernet transmit frame source address field. Set to force the MAC to automatically insert the Ethernet MAC source address into the Ethernet transmit frame source address. The SA1, SA2, and SA3 registers provide the address information. When the ITXA bit is cleared, the Ethernet MAC source address is taken from the data in the TX_FIFO.
D07	R/W	RXRAM	1	**RX FIFO RAM access** 0 CPU access to the RX FIFO RAM is disabled 1 CPU access to the RX FIFO RAM is enabled
D06:00	N/A	Reserved	N/A	N/A

Ethernet General Control Register #2

Address: A060 0004

Register



Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

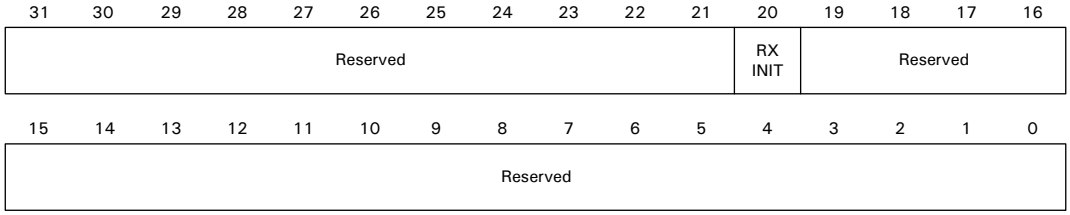
Bits	Access	Mnemonic	Reset	Description
D31:08	R/W	Not used	0	Always write as 0.
D07	R/W	TCLER	0	**Clear transmit error** 0→1 transition: Clear transmit error. Clears out conditions in the transmit packet processor that have caused the processor to stop and require assistance from software before the processor can be restarted (for example, an AHB bus error or the TXBUFNR bit set in the Ethernet Interrupt Status register). Toggle this bit from low to high to restart the transmit packet processor.
D06:04	R/W	Not used	0	**Always write as 0.**
D03	R/W	TKICK	0	**Transmit DMA state machine enable** 0→1 transition, used by software to start a DMA transfer after a buffer descriptor has been updated.
D02	R/W	AUTOZ	0	**Enable statistics counter clear on read** 0 No change in counter value after read 1 Counter cleared after read When set, configures all counters in the Statistics module to clear on read. If AUTOZ is not set, the counters retain their value after a read. The counters can be cleared by writing all zeros.
D01	R/W	CLRCNT	1	**Clear statistics counters** 0 Do not clear all counters 1 Clear all counters When set, synchronously clears all counters in the Statistics module.
D00	R/W	STEN	0	**Enable statistics counters** 0 Counters disabled 1 Counters enabled When set, enables all counters in the Statistics module. If this bit is cleared, the counters will not update.

Ethernet General Status register

Address: A060 0008

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

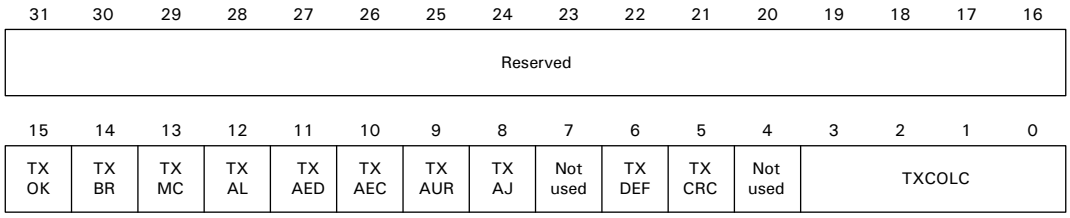
Bits	Access	Mnemonic	Reset	Description
D31:21	N/A	Reserved	N/A	N/A
D20	R/C	RXINIT	0x0	**RX initialization complete** Set when the RX_RD logic has completed the initialization of the local buffer descriptor registers requested when ERXINIT in Ethernet General Control Register #1 is set. The delay from ERXINIT set to RXINIT set is less than five microseconds.
D19:00	N/A	Reserved	N/A	N/A

Ethernet Transmit Status register

Address: A060 0018

The Ethernet Status register contains the status for the last transmit frame. The TXDONE bit in the Ethernet Interrupt Status register (see page 317) is set upon completion of a transmit frame and the Ethernet Transmit Status register is loaded at the same time. Bits [15:0] are also loaded into the Status field of the last transmit buffer descriptor for the frame.

Register



Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:16	N/A	Reserved	N/A	N/A
D15	R	TXOK	0x0	**Frame transmitted OK** When set, indicates that the frame has been delivered to and emptied from the transmit FIFO without problems.
D14	R	TXBR	0x0	**Broadcast frame transmitted** When set, indicates the frame's destination address was a broadcast address.
D13	R	TXMC	0x0	**Multicast frame transmitted** When set, indicates the frame's destination address was a multicast address.
D12	R	TXAL	0x0	**TX abort — late collision** When set, indicates that the frame was aborted due to a collision that occurred beyond the collision window set in the Collision Window/Retry register. If this bit is set, the TX_WR logic stops processing frames and sets the TXERR bit in the Ethernet Interrupt Status register.
D11	R	TXAED	0x0	**TX abort — excessive deferral** When set, indicates that the frame was deferred in excess of 6071 nibble times in 100 Mbps or 24,287 times in 0 Mbps mode. This causes the frame to be aborted if the *excessive deferral bit* is set to 0 in MAC Configuration Register #2. If TXAED is set, the TX_WR logic stops processing frames and sets the TXERR bit in the Ethernet Interrupt Status register.
D10	R	TXAEC	0x0	**TX abort — excessive collisions** When set, indicates that the frame was aborted because the number of collisions exceeded the value set in the Collision Window/Retry register. If this bit is set, the TX_WR logic stops processing frames and sets the TXERR bit in the Ethernet Interrupt Status register.
D09	R	TXAUR	0x0	**TX abort — underrun** When set, indicates that the frame was aborted because the TX_FIFO had an underrun. If this bit is set, the TX_WR logic stops processing frames and sets the TXERR bit in the Ethernet Interrupt Status register.

Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Reset	Description
D08	R	TXAJ	0x0	**TX abort — jumbo** When set, indicates that the frame's length exceeded the value set in the Maximum Frame register. TXAJ is set only if the HUGE bit in MAC Configuration Register #2 is set to 0. Jumbo frames result in the TX buffer descriptor buffer length field being set to 0x000. If the HUGE bit is set to 0, the frame is truncated. If TXAJ is set, the TX_WR logic stops processing frames and sets the TXERR bit in the Ethernet Interrupt Status register.
D07	R	Not used	0x0	Always set to 0.
D06	R	TXDEF	0x0	**Transmit frame deferred** When set, indicates that the frame was deferred for at least one attempt, but less than the maximum number for an excessive deferral. TXDEF is also set when a frame was deferred due to a collision. This bit is not set for late collisions.
D05	R	TXCRC	0x0	**Transmit CRC error** When set, indicates that the attached CRC in the frame did not match the internally-generated CRC. This bit is not set if the MAC is inserting the CRC in the frame (that is, the CRCEN bit is set in MAC Configuration Register #2). If TXCRC is set, the TX_WR logic stops processing frames and sets the TXERR bit in the Ethernet Interrupt Status register.
D04	R	Not used	0x0	Always set to 0.
D03:00	R	TXCOLC	0x0	**Transmit collision count** Number of collisions the frame incurred during transmission attempts.

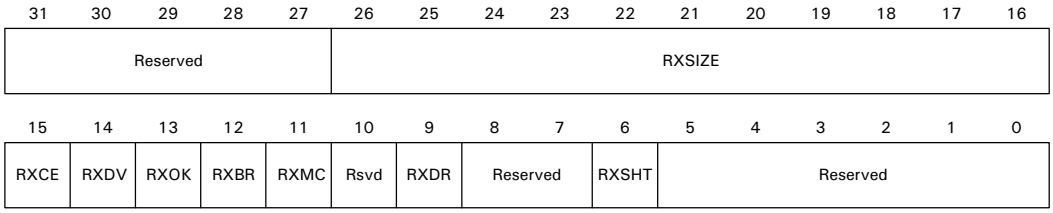
Ethernet Receive Status register

Address: A060 001C

The Ethernet Receive Status register contains the status for the last completed receive frame. The RXBR bit in the Ethernet Interrupt Status register (see page 317) is set whenever a receive frame is completed and the Ethernet Receive Status register is loaded at the same time. Bits [15:0] are also loaded into the status field of the receive buffer descriptor used for the frame.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:27	N/A	Reserved	N/A	N/A
D26:16	R	RXSIZE	0x000	**Receive frame size in bytes** Length of the received frame, in bytes.
D15	R	RXCE	0x0	**Receive carrier event previously seen** When set, indicates that a *carrier event* activity (an activity on the receive channel that does not result in a frame receive attempt being made) was found at some point since the last receive statistics. A carrier event results when the interface signals to the PHY have the following values: MRXER = 1 MRXDV = 0 RXD = 0xE The event is being reported with this frame, although it is not associated with the frame.
D14	R	RXDV	0x0	**Receive data violation event previously seen** Set when the last receive event was not long enough to be a valid frame.
D13	R	RXOK	0x0	**Receive frame OK** Set when the frame has a valid CRC and no symbol errors.
D12	R	RXBR	0x0	**Receive broadcast frame** Set when the frame has a valid broadcast address.
D11	R	RXMC	0x0	**Receive multicast frame** Set when the frame has a valid multicast address.
D10	N/A	Reserved	N/A	N/A
D09	R	RXDR	0x0	**Receive frame has dribble bits** Set when an additional 1–7 bits are received after the end of the frame.
D08:07	N/A	Reserved	N/A	N/A

Downloaded from www.Manualslib.com manuals search engine

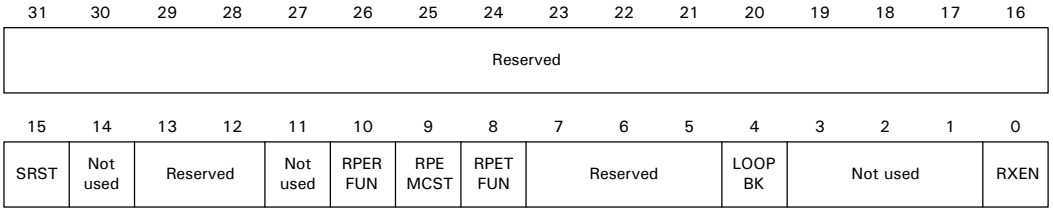
Bits	Access	Mnemonic	Reset	Description
D06	R	RXSHT	0x0	**Receive frame is too short** Set when the frame's length is less than 64 bytes. Short frames are accepted only when the ERXSHT bit is set to 1 in Ethernet General Control Register #1.
D05:00	N/A	Reserved	N/A	N/A

MAC Configuration Register #1

Address: A060 0400

MAC Configuration Register #1 provides bits that control functionality within the Ethernet MAC block.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:16	N/A	Reserved	N/A	N/A
D15	R/W	SRST	1	**Soft reset** Set this bit to 1 to reset the RX_WR, TX_RD, MAC (except host interface), SAL (except host interface).
D14	R/W	Not used	0	Always write as 0.
D13:12	N/A	Reserved	N/A	N/A
D11	R/W	Not used	0	Always write as 0.
D10	R/W	RPERFUN	0	**Reset PERFUN** Set this bit to 1 to put the MAC receive logic into reset.
D09	R/W	RPEMCST	0	**Reset PEMCS/TX** Set this bit to 1 to put the MAC control sublayer/transmit domain logic into reset.
D08	R/W	RPETFUN	0	**Reset PETFUN** Set this bit to 1 to put the MAC transmit logic into reset.
D07:05	N/A	Reserved	N/A	N/A

Downloaded from www.Manualslib.com manuals search engine

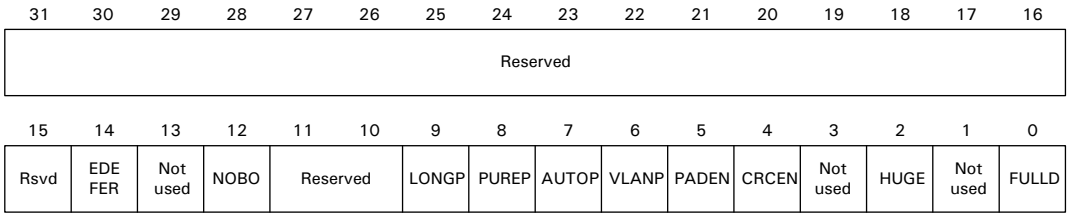
Bits	Access	Mnemonic	Reset	Description
D04	R/W	LOOPBK	0	**Internal loopback** Set this bit to 1 to cause the MAC transmit interface to be internally looped back to the MAC receive interface. Clearing this bit results in normal operation.
D03:01	R/W	Not used	0	Always write as 0.
D00	R/W	RXEN	0	**Receive enable** Set this bit to 1 to allow the MAC receiver to receive frames.

MAC Configuration Register #2

Address: A060 0404

MAC Configuration Register #2 provides additional bits that control functionality within the Ethernet MAC block.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Definition
D31:15	N/A	Reserved	N/A	N/A
D14	R/W	EDEFER	0	**Excess deferral** 0 The MAC aborts when the excessive deferral limit is reached (that is, 6071 nibble times in 100 Mbps mode or 24,287 bit times in 10 Mbps mode). 1 Enables the MAC to defer to carrier indefinitely, as per the 802.3u standard.
D13	R/W	Not used	0	Always write to 0.
D12	R/W	NOBO	0	**No backoff** When this bit is set to 1, the MAC immediately retransmits following a collision, rather than using the binary exponential backoff algorithm (as specified in the 802.3u standard).
D11:10	N/A	Reserved	N/A	N/A

Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Reset	Definition
D09	R/W	LONGP	0	**Long preamble enforcement** 0 Allows any length preamble (as defined in the 802.3u standard). 1 The MAC allows only receive frames that contain preamble fields less than 12 bytes in length.
D08	R/W	PUREP	0	**Pure preamble enforcement** 0 No preamble checking is performed 1 The MAC certifies the content of the preamble to ensure that it contains 0x55 and is error-free.
D07	R/W	AUTOP	0	**Auto detect pad enable** When set to 1, this bit causes the MAC to detect automatically the type of transmit frame, either tagged or untagged, by comparing the two octets following the source address with the 0x8100 VLAN protect ID and pad accordingly. Note: This bit is ignored if PADEN is set to 0. See "PAD operation table for transmit frames" below.
D06	R/W	VLANP	0	**VLAN pad enable** Set to 1 to have the MAC pad all short transmit frames to 64 bytes and to append a valid CRC. This bit is used in conjunction with auto detect pad enable (AUTOP) and pad/CRC enable (PADEN). See "PAD operation table for transmit frames" below. This bit is ignored if PADEN is set to 0.
D05	R/W	PADEN	0	**Pad/CRC enable** 0 Short transmit frames not padded. 1 The MAC pads all short transmit frames. This bit is used in conjunction with auto detect pad enable (AUTOP) and VLAN pad enable (VLANP). See "PAD operation table for transmit frames" below.
D04	R/W	CRCEN	0	**CRC enable** 0 Transmit frames presented to the MAC contain a CRC. 1 Append a CRC to every transmit frame, whether padding is required or not. CRCEN must be set if PADEN is set to 1.
D03	R/W	Not used	0	Always write as 0.
D02	R/W	HUGE	0	**Huge frame enable** 0 Transmit and receive frames are limited to the MAXF value in the Maximum Frame register. 1 Frames of any length are transmitted and received.

Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Reset	Definition
D01	R/W	Not used	0	Always write as 0.
D00	R/W	FULLD	0	**Full-duplex** 0 The MAC operates in half-duplex mode. 1 The MAC operates in full-duplex mode.

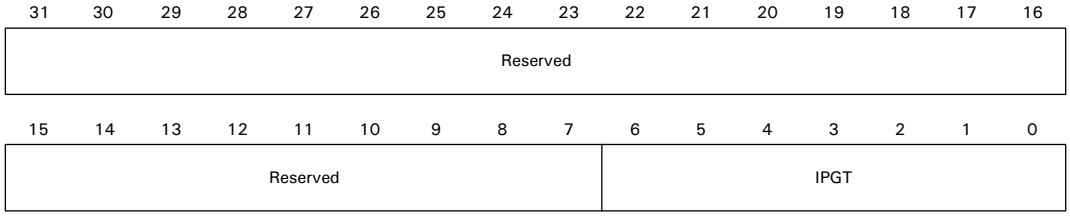
PAD operation table for transmit frames

Type	AUTOP	VLANP	PADEN	Action
Any	X	X	0	No pad; check CRC
Any	0	0	1	Pad to 60 bytes; append CRC
Any	X	1	1	Pad to 64 bytes; append CRC
Any	1	0	1	If untagged, pad to 60 bytes; append CRC If VLAN tagged, pad to 64 bytes; append CRC

Back-to-Back Inter-Packet-Gap register

Address: A060 0408

Register



Downloaded from www.Manualslib.com manuals search engine

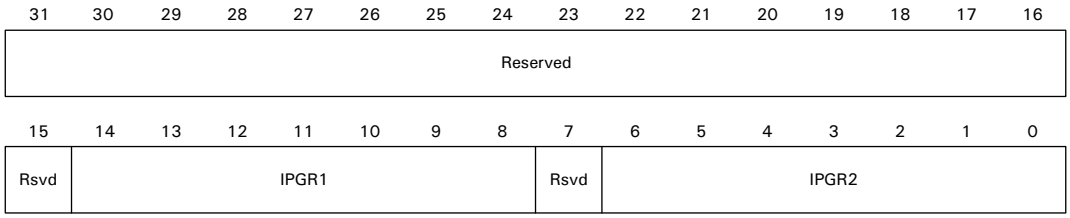
Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:07	N/A	Reserved	N/A	N/A
D06:00	R/W	IPGT	0x00	**Back-to-back inter-packet-gap** Programmable field that indicates the nibble time offset of the minimum period between the end of any transmitted frame to the beginning of the next frame. **Full-duplex mode** ■ Register value should be the appropriate period in nibble times minus 3. ■ Recommended setting is 0x15 (21d), which represents the minimum IPG of 0.96 uS (in 100 Mbps) or 9.6uS (in 10 Mbps). **Half-duplex mode** ■ Register value should be the appropriate period in nibble times minus 6. ■ Recommended setting is 0x12 (18d), which represents the minimum IPG of 0.96 uS (in 100 Mbps) or 9.6 uS (in 10 Mbps).

Non Back-to-Back Inter-Packet-Gap register

Address: A060 040C

Register



Downloaded from www.Manualslib.com manuals search engine

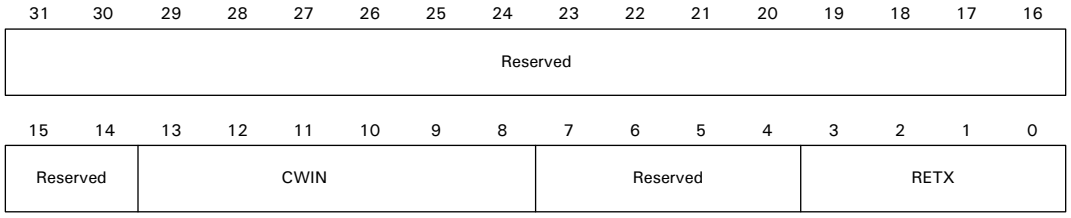
Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:15	N/A	Reserved	N/A	N/A
D14:08	R/W	IPGR1	0x00	**Non back-to-back inter-packet-gap part 1** Programmable field indicating optional carrierSense window (referenced in IEEE 8.2.3/4.2.3.2.1). ■ If carrier is detected during the timing of IPGR1, the MAC defers to carrier. ■ If carrier comes after IPGR1, the MAC continues timing IPGR2 and transmits — knowingly causing a collision. This ensures fair access to the medium. IPGR1's range of values is 0x0 to IPGR2. The recommended value is 0xC.
D07	N/A	Reserved	N/A	N/A
D06:00	R/W	IPGR2	0x00	**Non back-to-back inter-packet-gap part 2** Programmable field indicating the non back-to-back inter-packet-gap. The recommended value for this field is 0x12 (18d), which represents the minimum IPG of 0.96 µS in 100 Mbps or 9.6 µS in 10 Mbps.

Collision Window/Retry register

Address: A060 0410

Register



Downloaded from www.Manualslib.com manuals search engine

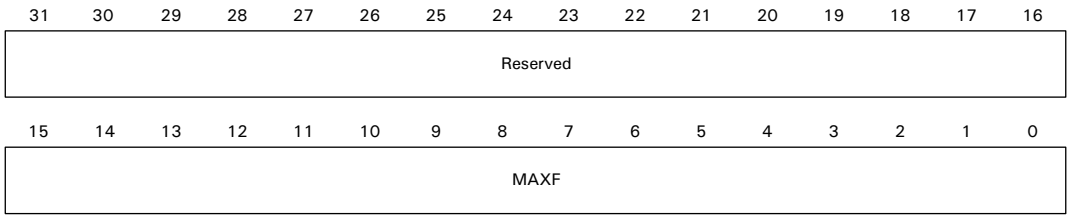
Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:14	N/A	Reserved	N/A	N/A
D13:08	R/W	CWIN	0x37	**Collision window** Programmable field indicating the slot time or collision window during which collisions occur in properly configured networks. Because the collision window starts at the beginning of transmissions, the preamble and SFD (start-of-frame delimiter) are included. The default value (0x37 (55d)) corresponds to the frame byte count at the end of the window.
D07:04	N/A	Reserved	N/A	N/A
D03:00	R/W	RETX	0xF	**Retransmission maximum** Programmable field specifying the number of retransmission attempts following a collision before aborting the frame due to excessive collisions. The 802.3u standard specifies the attemptLimit to be 0xF (15d).

Maximum Frame register

Address: A060 0414

Register



Downloaded from www.Manualslib.com manuals search engine

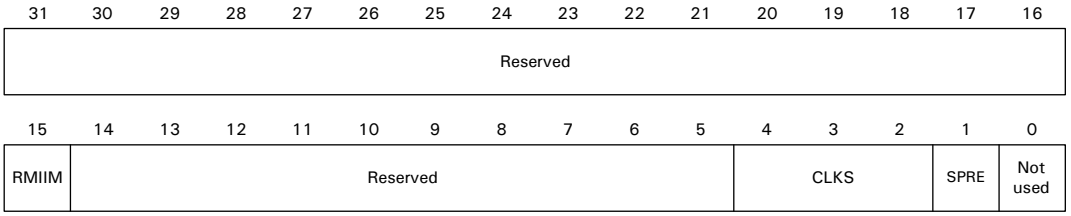
Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:16	N/A	Reserved	N/A	N/A
D15:00	R/W	MAXF	0x0600	**Maximum frame length** Default value of 0x600 represents a maximum receive frame of 1536 octets. An untagged maximum-size Ethernet frame is 1518 octets. A tagged frame adds four octets for a total of 1522 octets. To use a shorter maximum length restriction, program this field accordingly. Note: If a proprietary header is allowed, this field should be adjusted accordingly. For example, if 4-byte proprietary headers are prepended to the frames, the MAXF value should be set to 1526 octets. This allows the maximum VLAN tagged frame plus the 4-byte header.

MII Management Configuration register

Address: A060 0420

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:16	N/A	Reserved	N/A	N/A
D15	R/W	RMIIM	0	**Reset MII management block** Set this bit to 1 to reset the MII Management module.
D14:05	N/A	Reserved	N/A	N/A

Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Reset	Description
D04:02	R/W	CLKS	0x0	**Clock select** Used by the clock divide logic in creating the MII management clock, which (per the IEEE 802.3u standard) can be no faster than 2.5 MHz. Note: Some PHYs support clock rates up to 12.5 MHz. The AHB bus clock is used as the input to the clock divide logic. See the "Clocks field settings" table for settings that can be used with AHB clock (hclk) frequencies.
D01	R/W	SPRE	0	**Suppress preamble** 0 Causes normal cycles to be performed 1 Causes the MII Management module to perform read/write cycles without the 32-bit preamble field. (Preamble suppression is supported by some PHYs.)
D00	R/W	Not used	0	Always write to 0.

Clocks field settings

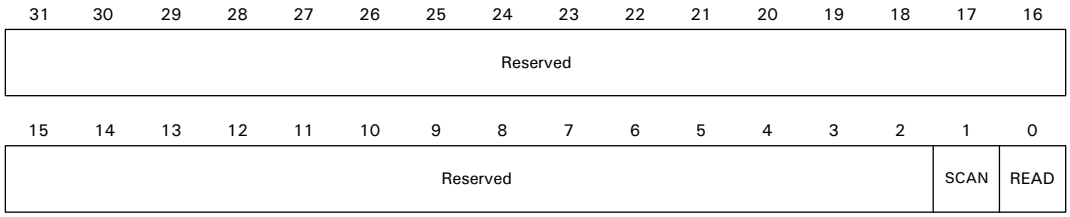
CLKS field	Divisor	AHB bus clock for 2.5 MHz (max) MII management clock	AHB bus clock for 12.5 MHz (max) MII management clock
000	4		
001	4		37.5 MHz
010	6		74.9 MHz
011	8		
100	10		
101	20	37.5 MHz	
110	30	74.9 MHz	
111	40		

MII Management Command register

Address: A060 0424

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

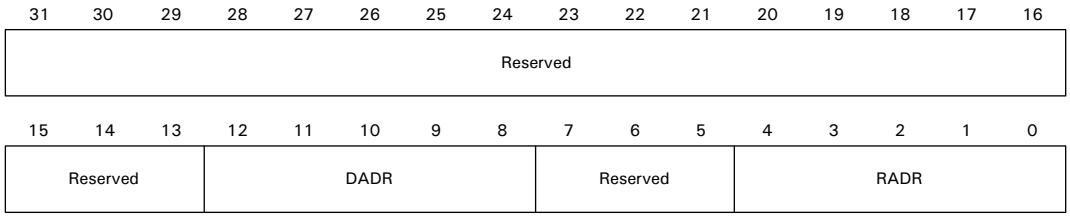
Note: If both SCAN and READ are set, SCAN takes precedence.

Bits	Access	Mnemonic	Reset	Description
D31:02	N/A	Reserved	N/A	N/A
D01	R/W	SCAN	0	**Automatically scan for read data** Set to 1 to have the MII Management module perform read cycles continuously. This is useful for monitoring link fail, for example. Note: SCAN must transition from a 0 to a 1 to initiate the continuous read cycles.
D00	R/W	READ	0	**Single scan for read data** Set to 1 to have the MII Management module perform a single read cycle. The read data is returned in the MII Management Read Data register after the BUSY bit in the MII Management Indicators register has returned to a value of 0. Note: READ must transition from a 0 to a 1 to initiate a single read cycle.

MII Management Address register

Address: A060 0428

Register



Downloaded from www.Manualslib.com manuals search engine

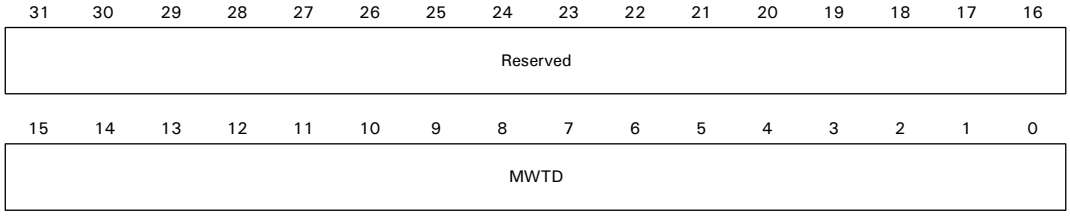
Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:13	N/A	Reserved	N/A	N/A
D12:08	R/W	DADR	0x00	**MII PHY device address** Represents the 5-bit PHY device address field for management cycles. Up to 32 different PHY devices can be addressed.
D07:05	N/A	Reserved	N/A	N/A
D04:00	R/W	RADR	0x00	**MII PHY register address** Represents the 5-bit PHY register address field for management cycles. Up to 32 registers within a single PHY device can be addressed.

MII Management Write Data register

Address: A060 042C

Register



Register bit assignment

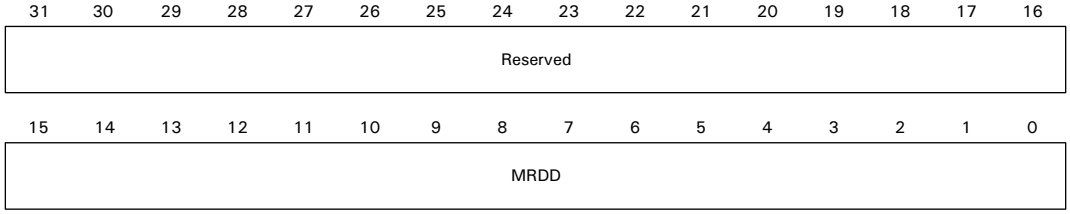
Bits	Access	Mnemonic	Reset	Description
D31:16	N/A	Reserved	N/A	N/A
D15:00	R/W	MWTD	0x0000	**MII write data** When this register is written, an MII Management write cycle is performed using this 16-bit data along with the preconfigured PHY device and PHY register addresses defined in the MII Management Address register. The write operation completes when the BUSY bit in the MII Management Indicators register returns to 0.

MII Management Read Data register

Address: A060 0430

Downloaded from www.Manualslib.com manuals search engine

Register



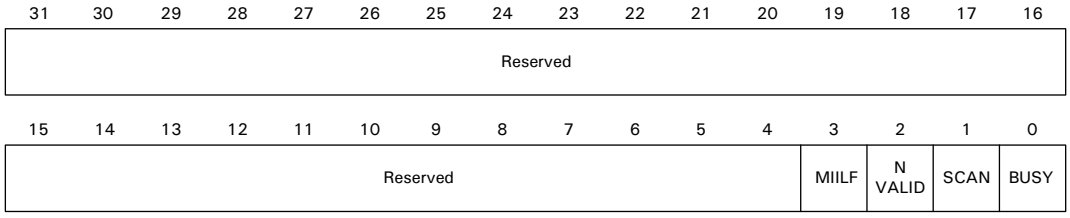
Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:16	N/A	Reserved	N/A	N/A
D15:00	R	MRDD	0x0000	**MII read data** Read data is obtained by reading from this register after an MII Management read cycle. An MII Management read cycle is executed by loading the MII Management Address register, then setting the READ bit to 1 in the MII Management Command register. Read data is available after the BUSY bit in the MII Management Indicators register returns to 0.

MII Management Indicators register

Address: A060 0434

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:04	N/A	Reserved	N/A	N/A
D03	R	MIILF	0	**MII link failure** When set to 1, indicates that the PHY currently has a link fail condition.

Downloaded from www.Manualslib.com manuals search engine

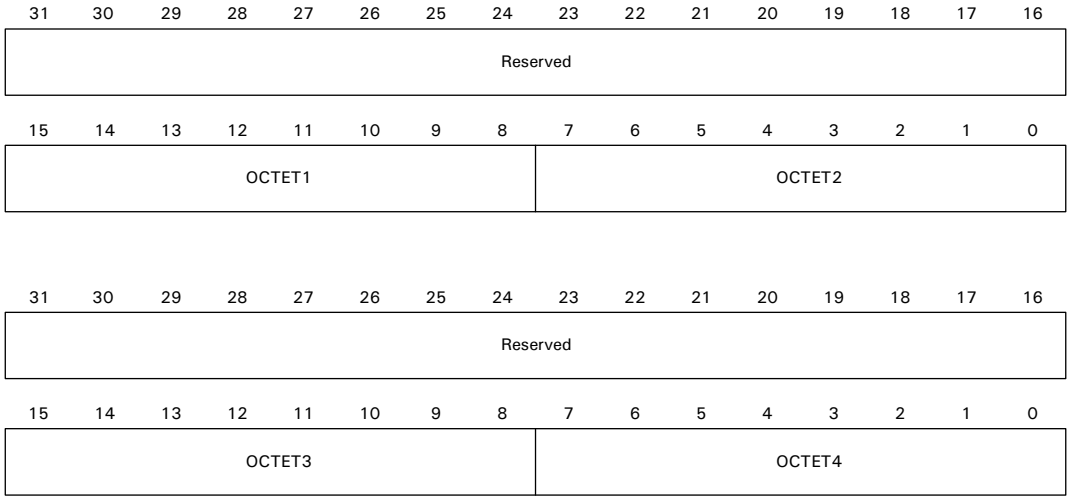
Bits	Access	Mnemonic	Reset	Description
D02	R	NVALID	0	**Read data not valid** When set to 1, indicates that the MII Management read cycle has not completed and the read data is not yet valid. Also indicates that SCAN READ is not valid for automatic scan reads.
D01	R	SCAN	0	**Automatically scan for read data in progress** When set to 1, indicates that continuous MII Management scanning read operations are in progress.
D00	R	BUSY	0	**MII interface BUSY with read/write operation** When set to 1, indicates that the MII Management module currently is performing an MII Management read or write cycle. This bit returns to 0 when the operation is complete.

Station Address registers

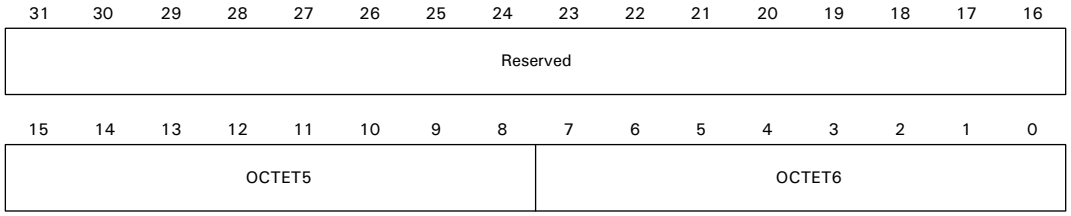
Addresses: A060 0440 / 0444 / 0448

The 48-bit station address is loaded into Station Address Register #1, Station Address Register #2, and Station Address Register #3, for use by the station address logic (see “Station address logic (SAL)” on page 264).

Registers



Downloaded from www.Manualslib.com manuals search engine



Register bit assignments for all three registers

Bits	Access	Mnemonic	Reset	Description
Station Address Register #1				
D31:16	N/A	Reserved	N/A	N/A
D15:08	R/W	OCTET1	0	Station address octet #1 (stad[7:0])
D07:00	R/W	OCTET2	0	Station address octet #2 (stad[15:8])
Station Address Register #2				
D31:16	N/A	Reserved	N/A	N/A
D15:08	R/W	OCTET3	0	Station address octet #3 (stad[23:16])
D07:00	R/W	OCTET4	0	Station address octet #4 (stad[31:24])
Station Address Register #3				
D31:16	N/A	Reserved	N/A	N/A
D15:08	R/W	OCTET5	0	Station address octet #5 (stad[39:32])
D07:00	R/W	OCTET6	0	Station address octet #6 (stad[47:40])

Note: Octet #6 is the first byte of a frame received from the MAC. Octet #1 is the last byte of the station address received from the MAC.

Station Address Filter register

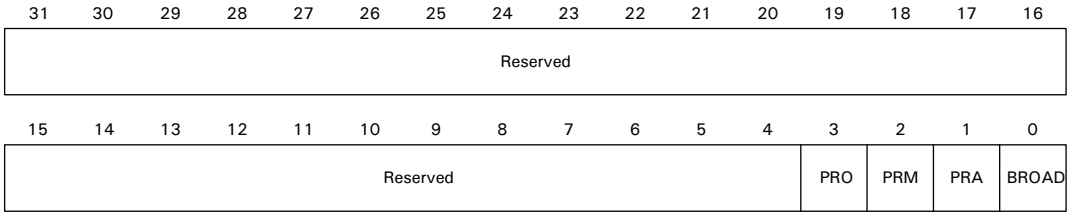
Address: A060 0500

The Station Address Filter register contains several filter controls. The register is located in the station address logic (see “Station address logic (SAL)” on page 264).

All filtering conditions are independent of each other. For example, the station address logic can be programmed to accept all multicast frames, all broadcast frames, and frames to the programmed destination address.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:04	N/A	Reserved	N/A	N/A
D03	R/W	PRO	0	Enable promiscuous mode; receive all frames
D02	R/W	PRM	0	Accept all multicast frames
D01	R/W	PRA	0	Accept multicast frames using the hash table
D00	R/W	BROAD	0	Accept all broadcast frames

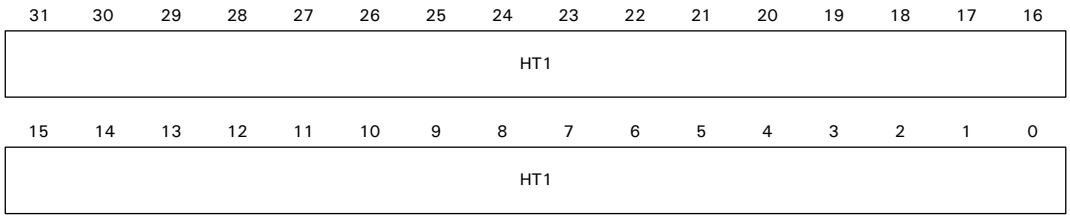
RegisterHash Tables

The MAC receiver provides the station address logic with a 6-bit CRC value that is the upper six bits of a 32-bit CRC calculation performed on the 48-bit multicast destination address. This 6-bit value addresses the 64-bit multicast hash table created in HT1 (hash table 1) and HT2 (hash table 2). If the current receive frame is a multicast frame and the 6-bit CRC addresses a bit in the hash table that is set to 1, the receive frame will be accepted; otherwise, the receive frame is rejected.

HT1 stores enables for the lower 32 CRC addresses; HT2 stores enables for the upper 32 CRC addresses.

HT1

Address: A060 0504



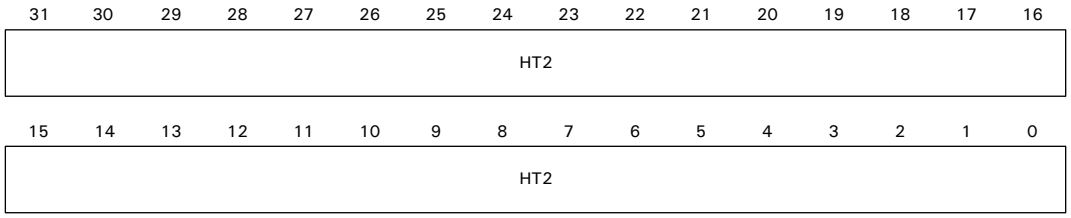
Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:00	R/W	HT1	0x00000000	CRC 31:00

Downloaded from www.Manualslib.com manuals search engine

HT2

Address: A060 0508



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:00	R/W	HT2	0x00000000	CRC 63:32

Statistics registers

Address: A060 0680 (base register)

The Statistics module has 39 counters and 4 support registers that count and save Ethernet statistics. The Ethernet General Control Register #2 contains three Statistics module configuration bits: AUTOZ, CLRCNT, and STEN. The counters support a “clear on read” capability that is enabled when AUTOZ is set to 1.

Combined transmit and receive statistics counters address map

The combined transmit and receive statistics counters are incremented for each good or bad frame, transmitted and received, that falls within the specified frame length limits of the counter (for example, TR127 counts 65–127 byte frames). The frame length excludes framing bits and includes the FCS (checksum) bytes. All counters are 18 bits, with this bit configuration:

D31:18	R		Reserved
D17:00	R/W	Reset = 0x00000	Count (R/W)

Address	Register	Transmit and receive counters			R/W
A060_0680	TR64	Transmit & receive 64		Byte frame counter	R/W
A060_0684	TR127	Transmit & receive 65	to	127 Byte frame counter	R/W
A060_0688	TR255	Transmit & receive 128	to	255 Byte frame counter	R/W
A060_068C	TR511	Transmit & receive 256	to	511 Byte frame counter	R/W
A060_0690	TR1K	Transmit & receive 512	to	1023 Byte frame counter	R/W

Downloaded from www.Manualslib.com manuals search engine

Address	Register	Transmit and receive counters			R/W
A060_0694	TRMAX	Transmit & receive 1024	to	1518 Byte frame counter	R/W
A060_0698	TRMGV	Transmit & receive 1519 count	to 1522 Byte good VLAN frame		R/W

Receive statistics counters address map

Address	Register	Receive counters	R/W
A060_069C	RBYT	Receive byte counter	R/W
A060_06A0	RPKT	Receive packet counter	R/W
A060_06A4	RFCS	Receive FCS error counter	R/W
A060_06A8	RMCA	Receive multicast packet counter	R/W
A060_06AC	RBCA	Receive broadcast packet counter	R/W
A060_06B0	RXCF	Receive control frame packet counter	R/W
A060_06B4	RXPF	Receive PAUSE frame packet counter	R/W
A060_06B8	RXUO	Receive unknown OPCODE counter	R/W
A060_06BC	RALN	Receive alignment error counter	R/W
A060_06C0	Reserved	N/A	N/A
A060_06C4	RCDE	Receive code error counter	R/W
A060_06C8	RCSE	Receive carrier sense error counter	R/W
A060_06CC	RUND	Receive undersize packet counter	R/W
A060_06D0	ROVR	Receive oversize packet counter	R/W
A060_06D4	RFRG	Receive fragments counter	R/W
A060_06D8	RJBR	Receive jabber counter	R/W
A060-06DC	Reserved	N/A	N/A

Receive byte counter (A060 069C)

Incremented by the byte count of frames received with 0 to 1518 bytes, including those in bad packets, excluding framing bits but including FCS bytes.

D31:24	R	Reset = Read as 0	Reserved
D23:00	R/W	Reset = 0x000000	RBYT

Receive packet counter (A060 06A0)

Incremented for each received frame (including bad packets, and all unicast, broadcast, and multicast packets).

D31:18	R	Reset = Read as 0	Reserved

Downloaded from www.Manualslib.com manuals search engine

D17:00	R/W	Reset = 0x00000	RPKT

Receive FCS error counter (A060 06A4)

Incremented for each frame received with a length of 64 to 1518 bytes, and containing a frame check sequence (FCS) error. FCS errors are not counted for VLAN frames that exceed 1518 bytes or for any frames with dribble bits.

D31:12	R	Reset = Read as 0	Reserved
D11:00	R/W	Reset = 0x000	RFCS

Receive multicast packet counter (A060 06A8)

Incremented for each good multicast frame with a length no greater than 1518 bytes (non-VLAN) or 1522 bytes (VLAN), excluding broadcast frames. This counter does not look at range/length errors.

D31:18	R	Reset = Read as 0	Reserved
D17:00	R/W	Reset = 0x00000	RMCA

Receive broadcast packet counter (A060 06AC)

Incremented for each good broadcast frame with a length no greater than 1518 bytes (non-VLAN) or 1522 bytes (VLAN), excluding multicast frames. This counter does not look at range/length errors.

D31:18	R	Reset = Read as 0	Reserved
D17:00	R/W	Reset = 0x00000	RBCA

Receive control frame packet counter (A060 06B0)

Incremented for each MAC control frame received (PAUSE and unsupported).

D31:12	R	Reset = Read as 0	Reserved
D11:00	R/W	Reset = 0x000	RXCF

Receive PAUSE frame packet counter (A060 06B4)

Incremented each time a valid PAUSE control frame is received.

D31:12	R	Reset = Read as 0	Reserved
D11:00	R/W	Reset = 0x000	RXPF

Receive unknown OPCODE packet counter (A060 06B8)

Incremented each time a MAC control frame is received with an OPCODE other than PAUSE.

D31:12	R	Reset = Read as 0	Reserved
D11:00	R/W	Reset = 0x000	RBUO

Downloaded from www.Manualslib.com manuals search engine

Receive alignment error counter (A060 06BC)

Incremented for each received frame, from 64 to 1518 bytes, that contains an invalid FCS and has dribble bits (that is, is not an integral number of bytes).

D31:12	R	Reset = Read as 0	Reserved
D11:00	R/W	Reset = 0x000	RALN

Receive code error counter (A060 06C4)

Incremented each time a valid carrier was present and at least one invalid data symbol was found.

D31:12	R	Reset = Read as 0	Reserved
D11:00	R/W	Reset = 0x000	RCDE

Receive carrier sense error counter (A060 06C8)

Incremented each time a false carrier is found during idle, as defined by a 1 on RX_ER and an 0xE on RXD. The event is reported with the statistics generated on the next received frame. Only one false carrier condition can be detected and logged between frames.

D31:12	R	Reset = Read as 0	Reserved
D11:00	R/W	Reset = 0x000	RCSE

Receive undersize packet counter (A060 06CC)

Incremented each time a frame is received that is less than 64 bytes in length, contains a valid FCS, and is otherwise well-formed. This counter does not look at range/length errors.

D31:12	R	Reset = Read as 0	Reserved
D11:00	R/W	Reset = 0x000	RUND

Receive oversize packet counter (A060 06D0)

Incremented each time a frame is received that exceeds 1518 bytes (non-VLAN) or 1522 bytes (VLAN), contains a valid FCS, and is otherwise well-formed. This counter does not look at range/length errors. This counter is not incremented when a packet is truncated because it exceeds the MAXF value.

D31:12	R	Reset = Read as 0	Reserved
D11:00	R/W	Reset = 0x000	ROVR

Receive fragments counter (A060 06D4)

Incremented for each frame received that is less than 64 bytes in length and contains an invalid FCS; this includes integral and non-integral lengths.

D31:12	R		Reserved
D11:00	R/W	Reset = 0x000	RFRG

Downloaded from www.Manualslib.com manuals search engine

Receive jabber counter (A060 06D8)

Incremented for frames received that exceed 1518 bytes (non-VLAN) or 1522 bytes (VLAN) and contain an invalid FCS, including alignment errors. This counter does not increment when a packet is truncated to 1518 (non-VLAN) or 1522 (VLAN) bytes by MAXF.

D31:12	R	Reset = Read as 0	Reserved
D11:00	R/W	Reset = 0x000	RJBR

Transmit statistics counters address map

Address	Register	Transmit counters	R/W
A060_06E0	TBYT	Transmit byte counter	R/W
A060_06E4	TPKT	Transmit packet counter	R/W
A060_06E8	TMCA	Transmit multicast packet counter	R/W
A060_06EC	TBCA	Transmit broadcast packet counter	R/W
A060_06F0	Reserved	N/A	N/A
A060_06F4	TDFR	Transmit deferral packet counter	R/W
A060_06F8	TEDF	Transmit excessive deferral packet counter	R/W
A060_06FC	TSCL	Transmit single collision packet counter	R/W
A060_0700	TMCL	Transmit multiple collision packet counter	R/W
A060_0704	TLCL	Transmit late collision packet counter	R/W
A060_0708	TXCL	Transmit excessive collision packet counter	R/W
A060_070C	TNCL	Transmit total collision counter	R/W
A060_0710	Reserved	N/A	N/A
A060_0714	Reserved	N/A	N/A
A060_0718	TJBR	Transmit jabber frame counter	R/W
A060_071C	TFCS	Transmit FCS error counter	R/W
A060_0720	Reserved	N/A	N/A
A060_0724	TOVR	Transmit oversize frame counter	R/W
A060_0728	TUND	Transmit undersize frame counter	R/W
A060_072C	TFRG	Transmit fragments frame counter	R/W

Transmit byte counter (A060 06E0)

Incremented by the number of bytes that were put on the wire, including fragments of frames that were involved with collisions. This count does not include preamble/SFD or jam bytes.

D31:24	R	Reset = Read as 0	Reserved
D23:00	R/W	Reset = 0x000000	TBYT

Downloaded from www.Manualslib.com manuals search engine

Transmit packet counter (A060 06E4)

Incremented for each transmitted packet (including bad packets, excessive deferred packets, excessive collision packets, late collision packets, and all unicast, broadcast, and multicast packets).

D31:18	R	Reset = Read as 0	Reserved
D17:00	R/W	Reset = 0x00000	TPKT

Transmit multicast packet counter (A060 06E8)

Incremented for each multicast valid frame transmitted (excluding broadcast frames).

D31:18	R	Reset = Read as 0	Reserved
D17:00	R/W	Reset = 0x00000	TMCA

Transmit broadcast packet counter (A060 06EC)

Incremented for each broadcast frame transmitted (excluding multicast frames).

D31:18	R	Reset = Read as 0	Reserved
D17:00	R/W	Reset = 0x00000	TBCA

Transmit deferral packet counter (A060 06F4)

Incremented for each frame that was deferred on its first transmission attempt. This counter does not include frames involved in collisions.

D31:12	R	Reset = Read as 0	Reserved
D11:00	R/W	Reset = 0x000	TDFR

Transmit excessive deferral packet counter (A060 06F8)

Incremented for frames aborted because they were deferred for an excessive period of time (3036 byte times).

D31:12	R	Reset = Read as 0	Reserved
D11:00	R/W	Reset = 0x000	TEDF

Transmit single collision packet counter (A060 06FC)

Incremented for each frame transmitted that experienced exactly one collision during transmission.

D31:12	R	Reset = Read as 0	Reserved
D11:00	R/W	Reset = 0x000	TSCL

Downloaded from www.Manualslib.com manuals search engine

Transmit multiple collision packet counter (A060 0700)

Incremented for each frame transmitted that experienced 2–15 collisions (including any late collisions) during transmission.

D31:12	R	Reset = Read as 0	Reserved
D11:00	R/W	Reset = 0x000	TMCL

Transmit late collision packet counter (A060 0704)

Incremented for each frame transmitted that experienced a late collision during a transmission attempt. Late collisions are defined using the CWIN[13:08] field of the Collision Window/Retry register.

D31:12	R	Reset = Read as 0	Reserved
D11:00	R/W	Reset = 0x000	TLCL

Transmit excessive collision packet counter (A060 0708)

Incremented for each frame transmitted that experienced excessive collisions during transmission, as defined by the RETX [03:00] field of the Collision Window/Retry register, and was aborted.

D31:12	R	Reset = Read as 0	Reserved
D11:00	R/W	Reset = 0x000	TXCL

Transmit total collision packet counter (A060 070C)

Incremented by the number of collisions experienced during the transmission of a frame.

Note: This register does not include collisions that result in an excessive collision count or late collisions.

D31:12	R	Reset = Read as 0	Reserved
D11:00	R/W	Reset = 0x000	TNCL

Transmit jabber frame counter (A060 0718)

Incremented for each oversized transmitted frame with an incorrect FCS value.

D31:12	R	Reset = Read as 0	Reserved
D11:00	R/W	Reset = 0x000	TJBR

Transmit FCS error counter (A060 071C)

Incremented for every valid-sized packet with an incorrect FCS value.

D31:12	R	Reset = Read as 0	Reserved
D11:00	R/W	Reset = 0x000	TFCS

Downloaded from www.Manualslib.com manuals search engine

Transmit oversize frame counter (A060 0724)

Incremented for each transmitted frame that exceeds 1518 bytes (NON_VLAN) or 1532 bytes (VLAN) and contains a valid FCS.

D31:12	R	Reset = Read as 0	Reserved
D11:00	R/W	Reset = 0x000	TOVR

Transmit undersize frame counter (A060 0728)

Incremented for every frame less than 64 bytes, with a correct FCS value. This counter also is incremented when a jumbo packet is aborted and the MAC is not checking the FCS, because the frame is reported as having a length of 0 bytes.

D31:12	R	Reset = Read as 0	Reserved
D11:00	R/W	Reset = 0x000	TUND

Transmit fragment counter (A060 072C)

Incremented for every frame less than 64 bytes, with an incorrect FCS value.

D31:12	R	Reset = Read as 0	Reserved
D11:00	R/W	Reset = 0x000	TFRG

General Statistics registers address map

These are the General Statistics registers.

Address	Register	General registers	R/W
A060_0730	CAR1	Carry Register 1	R
A060_0734	CAR2	Carry Register 2	R
A060_0738	CAM1	Carry Register 1 Mask register	R/W
A060_073C	CAM2	Carry Register 2 Mask register	R/W

Carry Register 1 (CAR1) and Carry Register 2 (CAR2) have carry bits for all of the statistics counters. These carry bits are set when the associated counter reaches a rollover condition.

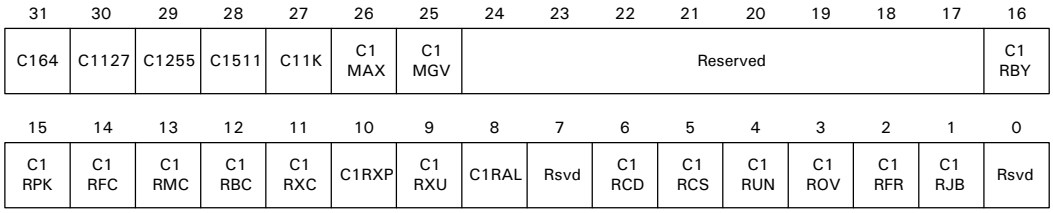
These carry bits also can cause the STOVFL (statistics counter overflow) bit in the Ethernet Interrupt Status register to be set. Carry Register 1 Mask register (CAM1) and Carry Register 2 Mask register (CAM2) have individual mask bits for each of the carry bits. When set, the mask bit prevents the associated carry bit from setting the STOVFL bit.

Carry Register 1

Address: A060 0730

Downloaded from www.Manualslib.com manuals search engine

Register


31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16
C164 C1127 C1255 C1511 C11K C1 MAX C1 MGV Reserved C1 RBY
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
C1 RPK C1 RFC C1 RMC C1 RBC C1 RXC C1RXP C1 RXU C1RAL Rsvd C1 RCD C1 RCS C1 RUN C1 ROV C1 RFR C1 RJB Rsvd

Register bit assignment

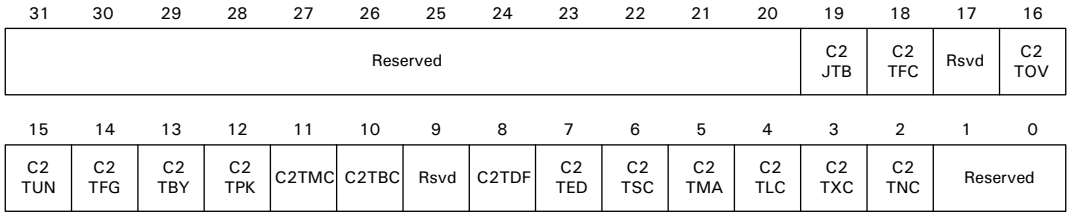
Bits	Access	Mnemonic	Reset	Description
D31	R/C	C164	0	Carry register 1 TR64 counter carry bit
D30	R/C	C1127	0	Carry register 1 TR127 counter carry bit
D29	R/C	C1255	0	Carry register 1 TR255 counter carry bit
D28	R/C	C1511	0	Carry register TR511 counter carry bit
D27	R/C	C11K	0	Carry register 1 TR1K counter carry bit
D26	R/C	C1MAX	0	Carry register 1 TRMAX counter carry bit
D25	R/C	C1MGV	0	Carry register 1 TRMGV counter carry bit
D24:17	N/A	Reserved	N/A	N/A
D16	R/C	C1RBY	0	Carry register 1 RBYT counter carry bit
D15	R/C	C1RPK	0	Carry register 1 RPKT counter carry bit
D14	R/C	C1RFC	0	Carry register 1 RFCS counter carry bit
D13	R/C	C1RMC	0	Carry register 1 RMCA counter carry bit
D12	R/C	C1RBC	0	Carry register 1 RBCA counter carry bit
D11	R/C	C1RXC	0	Carry register 1 RXCF counter carry bit
D10	R/C	C1RXP	0	Carry register 1 RXPF counter carry bit
D09	R/C	C1RXU	0	Carry register 1 RXUO counter carry bit
D08	R/C	C1RAL	0	Carry register 1 RALN counter carry bit
D07	N/A	Reserved	N/A	N/A
D06	R/C	C1RCD	0	Carry register 1 RCDE counter carry bit
D05	R/C	C1RCS	0	Carry register 1 RCSE counter carry bit
D04	R/C	C1RUN	0	Carry register 1 RUND counter carry register
D03	R/C	C1ROV	0	Carry register 1 ROVR counter carry bit
D02	R/C	C1RFR	0	Carry register 1 RFRG counter carry bit
D01	R/C	C1RJB	0	Carry register 1 RJBR counter carry bit
D00	N/A	Reserved	N/A	N/A

Carry Register 2

Address: A060 0734

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

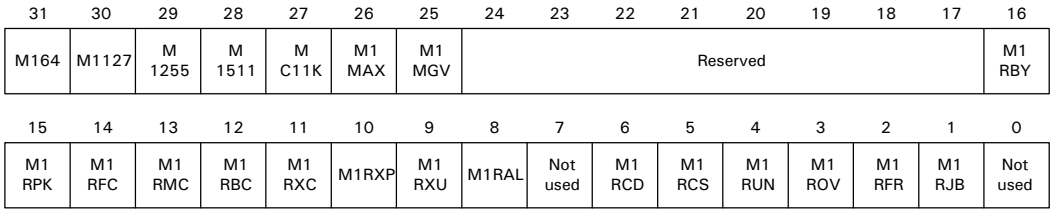
Bits	Access	Mnemonic	Reset	Description
D31:20	N/A	Reserved	N/A	N/A
D19	R/C	C2TJB	0	Carry register 2 TJBR counter carry bit
D18	R/C	C2TFC	0	Carry register 2 TFCS counter carry bit
D17	N/A	Reserved	N/A	N/A
D16	R/C	C2TOV	0	Carry register 2 TOVR counter carry bit
D15	R/C	C2TUN	0	Carry register 2 TUND counter carry bit
D14	R/C	C2TFG	0	Carry register 2 TFRG counter carry bit
D13	R/C	C2TBY	0	Carry register 2 TBYT counter carry bit
D12	R/C	C2TPK	0	Carry register 2 TPKT counter carry bit
D11	R/C	C2TMC	0	Carry register 2 TMCA counter carry bit
D10	R/C	C2TBC	0	Carry register 2 TBCA counter carry bit
D09	N/A	Reserved	N/A	N/A
D08	R/C	C2TDF	0	Carry register 2TDFR counter carry bit
D07	R/C	C2TED	0	Carry register 2 TEDF counter carry bit
D06	R/C	C2TSC	0	Carry register 2 TSCL counter carry bit
D05	R/C	C2TMA	0	Carry register 2 TMCL counter carry bit
D04	R/C	C2TLC	0	Carry register 2 TLCL counter carry bit
D03	R/C	C2TXC	0	Carry register 2 TXCL counter carry bit
D02	R/C	C2TNC	0	Carry register 2 TNCL counter carry bit
D01:00	N/A	Reserved	N/A	N/A

Carry Register 1 Mask register

Address: A060 0738

Downloaded from www.Manualslib.com manuals search engine

Register


31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16
M164 M1127 M 1255 M 1511 M C11K M1 MAX M1 MGV Reserved M1 RBY
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
M1 RPK M1 RFC M1 RMC M1 RBC M1 RXC M1RXP M1 RXU M1RAL Not used M1 RCD M1 RCS M1 RUN M1 ROV M1 RFR M1 RJB Not used

Register bit assignment

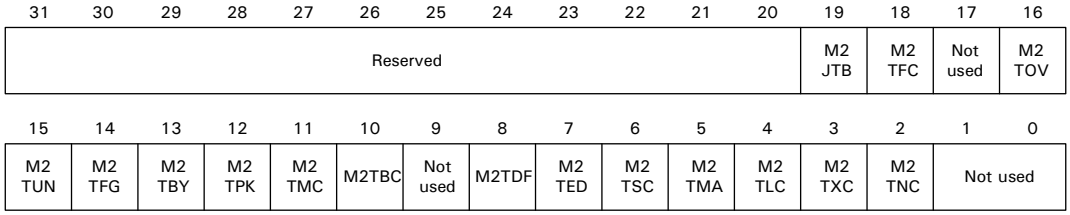
Bits	Access	Mnemonic	Reset	Description
D31	R/W	M164	1	Mask register 1 TR64 counter carry bit mask
D30	R/W	M1127	1	Mask register 1 TR127 counter carry bit mask
D29	R/W	M1255	1	Mask register 1 TR255 counter carry bit mask
D28	R/W	M1511	1	Mask register 1 TR511 counter carry bit mask
D27	R/W	M11K	1	Mask register 1 TR1K counter carry bit mask
D26	R/W	M1MAX	1	Mask register 1 TRMAX counter carry bit mask
D25	R/W	M1MGV	1	Mask register 1 TRMGV counter carry bit mask
D24:17	N/A	Reserved	N/A	N/A
D16	R/W	M1RBY	1	Mask register 1 RBYT counter carry bit mask
D15	R/W	M1RPK	1	Mask register 1 RPKT counter carry bit mask
D14	R/W	M1RFC	1	Mask register 1 RFCS counter carry bit mask. \
D13	R/W	M1RMC	1	Mask register 1 RMCA counter carry bit mask
D12	R/W	M1RBC	1	Mask register 1 RBCA counter carry bit mask
D11	R/W	M1RXC	1	Mask register 1 RXCF counter carry bit mask
D10	R/W	M1RXP	1	Mask register 1 RXPF counter carry bit mask
D09	R/W	M1RXU	1	Mask register 1 RXUO counter carry bit mask
D08	R/W	M1RAL	1	Mask register 1 RALN counter carry bit mask.
D07	R/W	Not used	1	Always write as 1.
D06	R/W	M1RCD	1	Mask register 1 RCDE counter carry bit mask
D05	R/W	M1RCS	1	Mask register 1 RCSE counter carry bit mask
D04	R/W	M1RUN	1	Mask register 1 RUND counter carry bit mask
D03	R/W	M1ROV	1	Mask register 1 ROVR counter carry bit mask
D02	R/W	M1RFR	1	Mask register 1 RFRG counter carry bit mask
D01	R/W	M1RJB	1	Mask register 1 RJBR counter carry bit mask
D00	R/W	Not used	1	Always write as 1.

Downloaded from www.Manualslib.com manuals search engine

Carry Register 2 Mask register

Address: A060 073C

Register



Register bit assignment

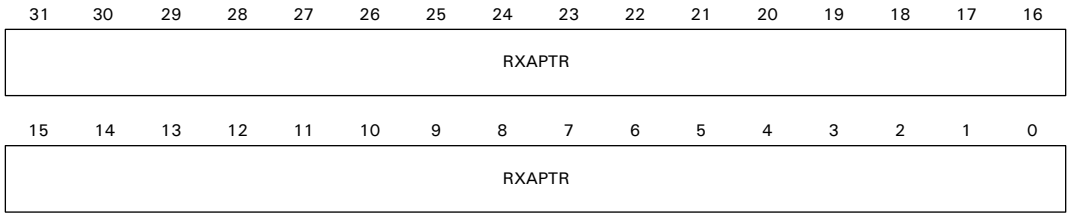
Bits	Access	Mnemonic	Reset	Description
D31:20	N/A	Reserved	N/A	N/A
D19	R/W	M2TJB	1	Mask register 2 TJBR counter carry bit mask
D18	R/W	M2TFC	1	Mask register 2 TFCS counter carry bit mask
D17	R/W	Not used	1	Always write as 1.
D16	R/W	M2TOV	1	Mask register 2 TOVR counter carry bit mask
D15	R/W	M2TUN	1	Mask register 2 TUND counter carry bit mask
D14	R/W	M2TFG	1	Mask register 2 TFRG counter carry bit mask
D13	R/W	M2TBY	1	Mask register 2 TBYT counter carry bit mask
D12	R/W	M2TPK	1	Mask register 2 TPKT counter carry bit mask
D11	R/W	M2TMC	1	Mask register 2 TMCA counter carry bit mask
D10	R/W	M2TBC	1	Mask register 2 TBCA counter carry bit mask
D09	R/W	Not used	1	Always write as 1.
D08	R/W	M2TDF	1	Mask register 2 TDFR counter carry bit mask
D07	R/W	M2TED	1	Mask register 2 TEDF counter carry bit mask
D06	R/W	M2TSC	1	Mask register 2 TSCL counter carry bit mask
D05	R/W	M2TMA	1	Mask register 2 TMCL counter carry bit mask
D04	R/W	M2TLC	1	Mask register 2 TLCL counter carry bit mask
D03	R/W	M2TXC	1	Mask register 2 TXCL counter carry bit mask
D02	R/W	M2TNC	1	Mask register 2 TNCL counter carry bit mask
D01:00	R/W	Not used	11	Always write as “11.”

Downloaded from www.Manualslib.com manuals search engine

RX_A Buffer Descriptor Pointer register

Address: A060 0A00

Register



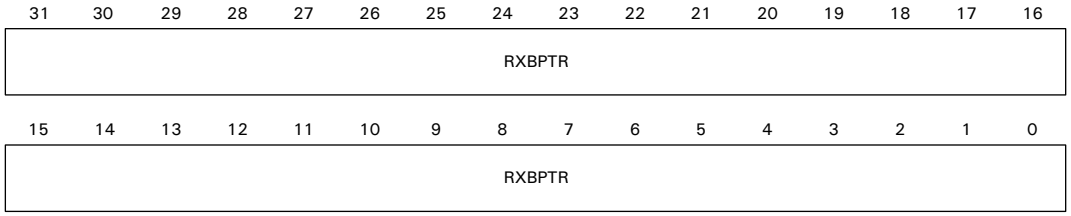
Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:00	R/W	RXAPTR	0x00000000	**RX_A Buffer Descriptor Pointer** Contains a pointer to the initial receive buffer descriptor for the A pool of buffers.

RX_B Buffer Descriptor Pointer register

Address: A060 0A04

Register



Register bit assignment

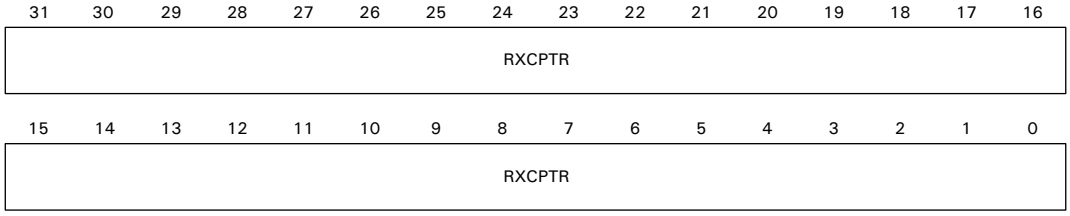
Bits	Access	Mnemonic	Reset	Description
D31:00	R/W	RXBPTR	0x00000000	**RX_B Buffer Descriptor Pointer** Contains a pointer to the initial receive buffer descriptor for the B pool of buffers.

Downloaded from www.Manualslib.com manuals search engine

RX_C Buffer Descriptor Pointer register

Address: A060 0A08

Register



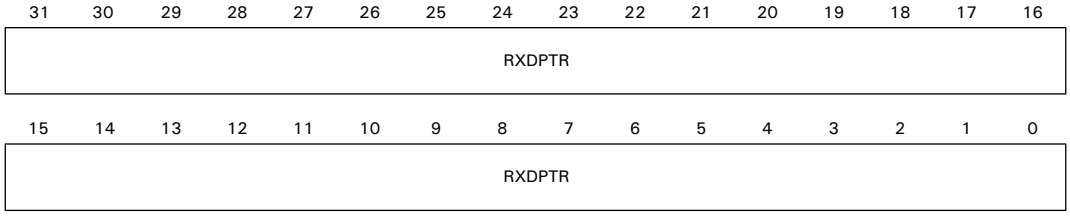
Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:00	R/W	RXCPTR	0x00000000	**RX_C Buffer Descriptor Pointer** Contains a pointer to the initial receive buffer descriptor for the C pool of buffers.

RX_D Buffer Descriptor Pointer register

Address: A060 0A0C

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:00	R/W	RXDPTR	0x00000000	**RX_D Buffer Descriptor Pointer** Contains a pointer to the initial receive buffer descriptor for the D pool of buffers.

Downloaded from www.Manualslib.com manuals search engine

Ethernet Interrupt Status register

Address: A060 0A10

The Ethernet Interrupt Status register contains status bits for all of the Ethernet interrupt sources. Each interrupt status bit is assigned to either the RX or TX Ethernet interrupt; bits D25:16 are assigned to the RX interrupt and D06:00 are assigned to the TX interrupt.

The bits are set to indicate an interrupt condition, and are cleared by writing a 1 to the appropriate bit. All interrupts bits are enabled using the Ethernet Interrupt Enable register (EINTREN). If any enabled bit in the Ethernet Interrupt Status register is set, its associated Ethernet interrupt to the system is set. The interrupt to the system is negated when all active interrupt sources have been cleared. If an interrupt source is active at the same time the interrupt bit is being cleared, the interrupt status bit remains set and the interrupt signal remains set.

Note: For diagnostics, software can cause any of these interrupt status bits to be set by writing a 1 to a bit that is 0.

Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						RX OVFL_ DATA	RX OVFL_ STAT	RX BUFC	RX DONE A	RX DONE B	RX DONE C	RX DONE D	RXNO BUF	RX BU FFUL	RXBR

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									ST OVFL	Not used	TX BUFC	TX BUF NR	TX DONE	TX ERR	TX IDLE

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:26	N/A	Reserved	N/A	N/A
D25	R/C	RXOVFL_DATA	0	Assigned to RX interrupt. RX data FIFO overflowed. For proper operation, reset the receive packet processor using the ERX bit in the Ethernet General Control Register #1 when an overflow condition occurs.
D24	R/C	RXOVFL_STAT	0	Assigned to RX interrupt. RX status FIFO overflowed.
D23	R/C	RXBUFC	0	Assigned to RX interrupt. I bit set in receive Buffer Descriptor and buffer closed.
D22	R/C	RXDONEA	0	Assigned to RX interrupt. Complete receive frame stored in pool A of system memory.

Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Reset	Description
D21	R/C	RXDONEB	0	Assigned to RX interrupt. Complete receive frame stored in pool B of system memory.
D20	R/C	RXDONEC	0	Assigned to RX interrupt. Complete receive frame stored in pool C of system memory.
D19	R/C	RXDONED	0	Assigned to RX interrupt. Complete receive frame stored in pool D of system memory.
D18	R/C	RXNOBUF	0	Assigned to RX interrupt. No buffer is available for this frame due to one of these conditions: ■ All four buffer rings being disabled ■ All four buffer rings being full ■ No available buffer big enough for the frame
D17	R/C	RXBUFFUL	0	Assigned to RX interrupt. No buffer is available for this frame because all four buffer rings are disabled or full.
D16	R/C	RXBR	0	Assigned to RX interrupt. New frame available in the RX_FIFO. This bit is used for diagnostics.
D15:07	N/A	Reserved	N/A	N/A
D06	R/C	STOVFL	0	Assigned to TX interrupt. Statistics counter overflow. Individual counters can be masked using the Carry Register 1 and 2 Mask registers. The source of this interrupt is cleared by clearing the counter that overflowed, and by clearing the associated carry bit in either Carry Register 1 or Carry Register 2 by writing a 1 to the bit.
D05	R	Not used	0	Always write as 0.
D04	R/C	TXBUFC	0	Assigned to TX interrupt. I bit set in the Transmit Buffer Descriptor and buffer closed.
D03	R/C	TXBUFNR	0	Assigned to TX interrupt. F bit not set in the Transmit Buffer Descriptor when read from the TX Buffer descriptor RAM.
D02	R/C	TXDONE	0	Assigned to TX interrupt. Frame transmission complete.

Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Reset	Description
D01	R/C	TXERR	0	Last frame not transmitted successfully. Assigned to TX interrupt. See “Ethernet Interrupt Status register” on page 317 for information about restarting the transmitter when this bit is set.
D00	R/C	TXIDLE	0	TX_WR logic has no frame to transmit. Assigned to TX interrupt. See “Ethernet Interrupt Status register” on page 317 for information about restarting the transmitter when this bit is set.

Ethernet Interrupt Enable register

Address: A060 0A14

The Ethernet Interrupt Enable register contains individual enable bits for each of the bits in the Ethernet Interrupt Status register. When these bits are cleared, the corresponding bit in the Ethernet Interrupt Status register cannot cause the interrupt signal to the system to be asserted when it is set.

Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						EN_RX OVFL_ DATA	EN_RX OVFL_ STAT	EN_ RX BUFC	EN_RX DONE A	EN_RX DONE B	EN_RX DONE C	EN_RX DONE D	EN_ RXNO BUF	EN_RX BUF FUL	EN_ RXBR

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									EN_ST OVFL	Not used	EN_TX BUFC	EN_TX BUF NR	EN_ TX DONE	EN_ TX ERR	EN_ TX IDLE

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:26	N/A	Reserved	N/A	N/A
D25	R/W	EN_RXOVFL_DATA	0	Enable the RXOVFL_DATA interrupt bit.
D24	R/W	EN_RXOVFL_STAT	0	Enable the RXOVFL_STATUS interrupt bit.
D23	R/W	EN_RXBUFC	0	Enable the RXBUFC interrupt bit.
D22	R/W	EN_RXDONEA	0	Enable the RXDONEA interrupt bit.
D21	R/W	EN_RXDONEB	0	Enable the RXDONEB interrupt bit.
D20	R/W	EN_RXDONEC	0	Enable the RXDONEC interrupt bit.
D19	R/W	EN_RXDONED	0	Enable the RXDONED interrupt bit.
D18	R/W	EN_RXNOBUF	0	Enable the RXNOBUF interrupt bit.
D17	R/W	EN_RXBUFFUL	0	Enable the RXBUFFUL interrupt bit.

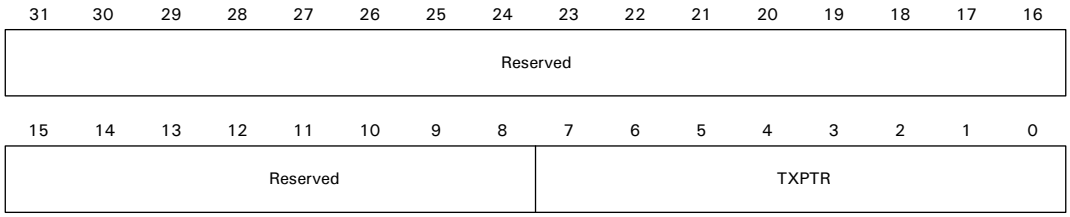
Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Reset	Description
D16	R/W	EN_RXBR	0	Enable the RXBR interrupt bit.
D15:07	N/A	Reserved	N/A	N/A
D06	R/W	EN_STOVFL	0	Enable the STOVFL interrupt bit.
D05	R/W	Not used	0	Always write as 0.
D04	R/W	EN_TXBUFC	0	Enable the TXBUFC interrupt bit.
D03	R/W	EN_TXBUFNR	0	Enable the TXBUFNR interrupt bit.
D02	R/W	EN_TXDONE	0	Enable the TXDONE interrupt bit.
D01	R/W	EN_TXERR	0	Enable the TXERR interrupt bit.
D00	R/W	EN_TXIDLE	0	Enable the TXIDLE interrupt bit.

TX Buffer Descriptor Pointer register

Address: A060 0A18

Register



Register bit assignment

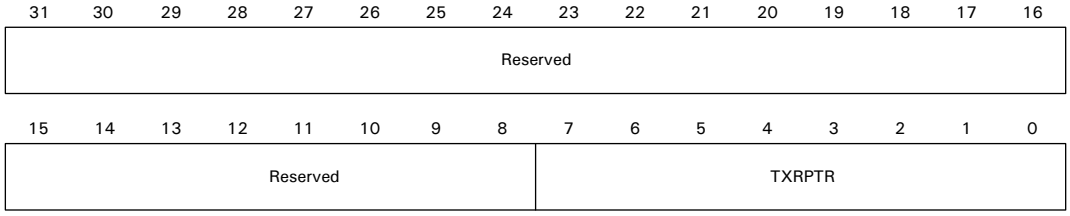
Bits	Access	Mnemonic	Reset	Description
D31:08	N/A	Reserved	N/A	N/A
D07:00	R/W	TXPTR	0x00	Contains a pointer to the initial transmit buffer descriptor in the TX buffer descriptor RAM. Note: This pointer is the 8-bit physical address of the TX buffer descriptor RAM, and points to the first location of the four-location buffer descriptor. The byte offset of this buffer descriptor can be calculated by multiplying this value by 4.

Downloaded from www.Manualslib.com manuals search engine

Transmit Recover Buffer Descriptor Pointer register

Address: A060 0A1C

Register



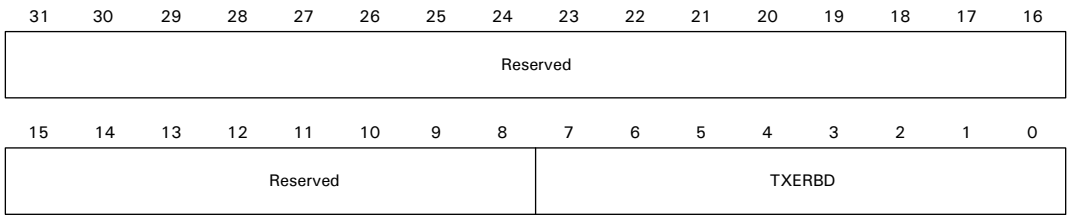
Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:08	N/A	Reserved	N/A	N/A
D07:00	R/W	TXRPTR	0x00	Contains a pointer to a buffer descriptor in the TX buffer descriptor RAM. Note: This pointer is the 8-bit physical address of the TX buffer descriptor RAM, and points to the first location of the four-location buffer descriptor. The byte offset of this buffer descriptor can be calculated by multiplying this value by 4. This is the buffer descriptor at which the TX_WR logic resumes processing when TCLER is toggled from low to high in Ethernet General Control Register #2.

TX Error Buffer Descriptor Pointer register

Address: A060 0A20

Register



Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:08	N/A	Reserved	N/A	N/A
D07:00	R	TXERBD	0x00	Contains the pointer (in the TX buffer descriptor RAM) to the last buffer descriptor of a frame that was not successfully transmitted. TXERBD is loaded by the TX_WR logic when a transmit frame is aborted by the MAC or when the MAC finds a CRC error in a frame. TXERBD also is loaded if a buffer descriptor that is not the first buffer descriptor in a frame does not have its F bit set. Note: This pointer is the 8-bit physical address of the TX buffer descriptor RAM, and points to the first location of the four-location buffer descriptor. The byte offset of this buffer descriptor can be calculated by multiplying this value by 4. Note: Software uses TXERBD to identify frames that were not transmitted successfully.

TX Stall Buffer Descriptor Pointer register

Address: A060 0A24

Register

Downloaded from www.Manualslib.com manuals search engine

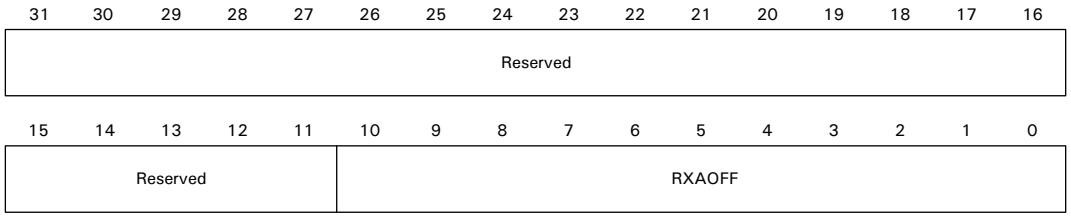
Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:08	N/A	Reserved	N/A	N/A
D07:00	R	TXSPTR	0x00	If the TX runs out of frames to send, it sets TXIDLE in the Ethernet Interrupt Status register and stores the pointer (in the TX buffer descriptor RAM) to the buffer descriptor that did not have its F bit set in the TX Stall Buffer Descriptor Pointer register. **Note:** This pointer is the 8-bit physical address of the TX buffer descriptor RAM, and points to the first location of the four-location buffer descriptor. The byte offset of this buffer descriptor can be calculated by multiplying this value by 4. Note: Software uses TXSPTR to identify the entry in the TX buffer descriptor RAM at which the TX stalled.

RX_A Buffer Descriptor Pointer Offset register

Address: A060 0A28

Register



Register bit assignment

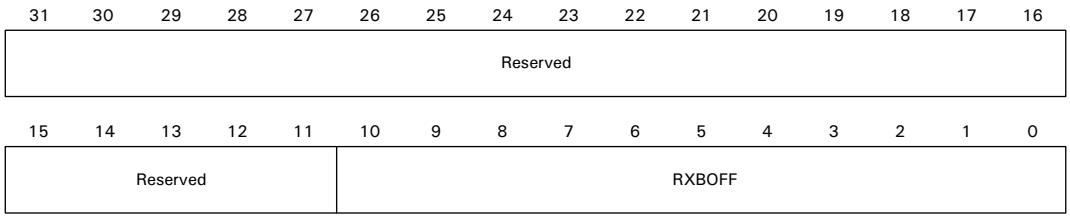
Bits	Access	Mnemonic	Reset	Description
D31:11	N/A	Reserved	N/A	N/A
D10:00	R	RXAOFF	0x000	Contains an 11-bit byte offset from the start of the pool A ring. The offset is updated at the end of the RX packet, and will have the offset to the next buffer descriptor that will be used. RXAOFF can be used to determine where the RX_RD logic will put the next packet.

Downloaded from www.Manualslib.com manuals search engine

RX_B Buffer Descriptor Pointer Offset register

Address: A060 0A2C

Register



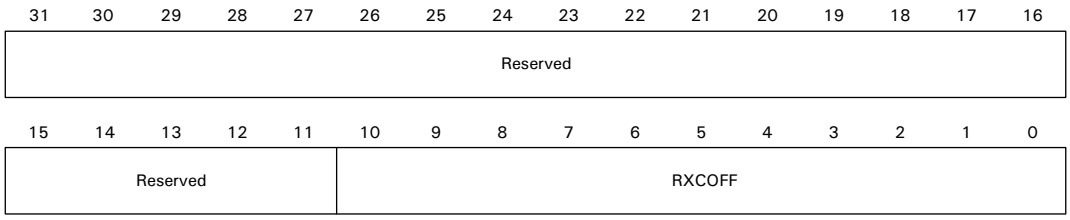
Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:11	N/A	Reserved	N/A	N/A
D10:00	R	RXBOFF	0x000	Contains an 11-bit byte offset from the start of the pool B ring. The offset is updated at the end of the RX packet, and will have the offset to the next buffer descriptor that will be used. RXBOFF can be used to determine where the RX_RD logic will put the next packet.

RX_C Buffer Descriptor Pointer Offset register

Address: A060 0A30

Register



Downloaded from www.Manualslib.com manuals search engine

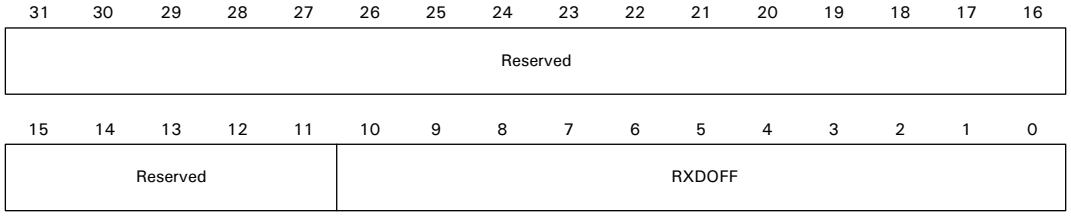
Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:11	N/A	Reserved	N/A	N/A
D10:00	R	RXCOFF	0x000	Contains an 11-bit byte offset from the start of the pool C ring. The offset is updated at the end of the RX packet, and will have the offset to the next buffer descriptor that will be used. RXCOFF can be used to determine where the RX_RD logic will put the next packet.

RX_D Buffer Descriptor Pointer Offset register

Address: A060 0A34

Register



Register bit assignment

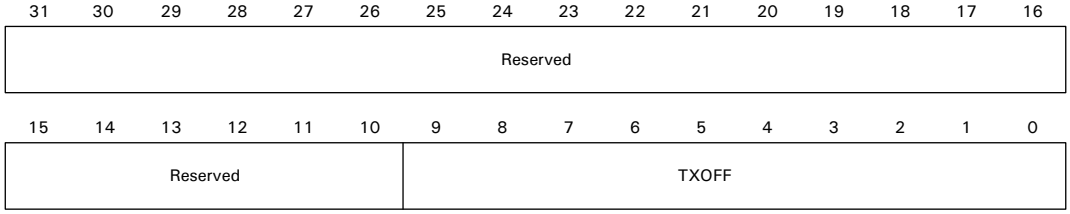
Bits	Access	Mnemonic	Reset	Description
D31:11	N/A	Reserved	N/A	N/A
D10:00	R	RXDOFF	0x000	Contains an 11-bit byte offset from the start of the pool D ring. The offset is updated at the end of the RX packet, and will have the offset to the next buffer descriptor that will be used. RXDOFF can be used to determine where the RX_RD logic will put the next packet.

Transmit Buffer Descriptor Pointer Offset register

Address: A060 0A38

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

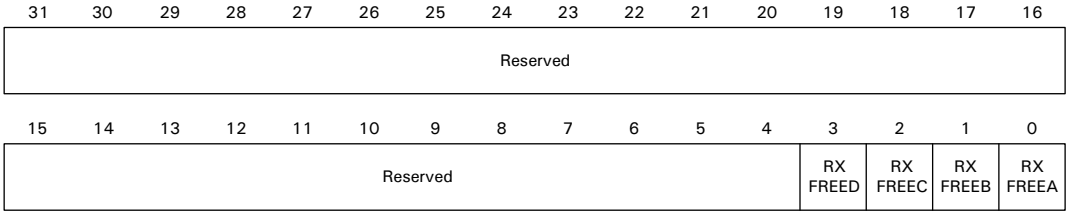
Bits	Access	Mnemonic	Reset	Description
D31:10	N/A	Reserved	N/A	N/A
D09:00	R	TXOFF	0x000	Contains a 10-bit byte offset from the start of the transmit ring in the TX buffer descriptor RAM. The offset is updated at the end of the TX packet, and will have the offset to the next buffer descriptor that will be used. TXOFF can be used to determine from where the TX_WR logic will grab the next packet.

RX Free Buffer register

Address: A060 0A3C

So the RX_RD logic knows when the software is freeing a buffer for reuse, the software writes to the RXFREE register each time it frees a buffer in one of the pools. RXFREE has an individual bit for each pool; this bit is set to 1 when the register is written. Reads to RXFREE always return all 0s.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:04	N/A	Reserved	N/A	N/A
D03	W	RXFREED	0	Pool D free bit
D02	W	RXFREEC	0	Pool C free bit

Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Reset	Description
D01	W	RXFREEB	0	Pool B free bit
D00	W	RXFREEA	0	Pool A free bit

Multicast Address Filter registers

Each of the eight entries in the multicast address filter logic has individual registers to hold its 48-bit multicast address. The multicast address for each entry is split between two registers. Each entry has a register that contains the lower 32 bits of the multicast address and a separate register that contains the upper 16 bits of the address. For an explanation of the synchronization scheme used for these registers, see “Clock synchronization” on page 276.

Multicast Low Address Filter Register #0

Address: A060 0A40

D31:00	R/W	Default = 0x0000 0000	MFILTL0

Multicast Low Address Filter Register #1

Address: A060 0A44

D31:00	R/W	Default = 0x0000 0000	MFILTL1

Multicast Low Address Filter Register #2

Address: A060 0A48

D31:00	R/W	Default = 0x0000 0000	MFILTL2

Multicast Low Address Filter Register #3

Address: A060 0A4C

D31:00	R/W	Default = 0x0000 0000	MFILTL4

Multicast Low Address Filter Register #4

Address: A060 0A50

D31:00	R/W	Default = 0x0000 0000	MFILTL4

Multicast Low Address Filter Register #5

Address: A060 0A54

D31:0	R/W	Default = 0x0000 0000	MFILTL5

Downloaded from www.Manualslib.com manuals search engine

Multicast Low Address Filter Register #6

Address: A060 0A58

D31:00	R/W	Default = 0x0000 0000	MFILTL6

Multicast Low Address Filter Register #7

Address: A060 0A5C

D31:00	R/W	Default = 0x0000 0000	MFILTL7

Multicast High Address Filter Register #0

Address: A060 0A60

D31:16	R	Default = 0x0000 0000	Reserved (read as 0)
D15:00	R/W	Default = 0x0000 0000	MFILTH0

Multicast High Address Filter Register #1

Address: A060 0A64

D31:16	R	Default = 0x0000 0000	Reserved (read as 0)
D15:00	R/W	Default = 0x0000 0000	MFILTH1

Multicast High Address Filter Register #2

Address: A060 0A68

D31:16	R	Default = 0x0000 0000	Reserved (read as 0)
D15:00	R/W	Default = 0x0000 0000	MFILTH2

Multicast High Address Filter Register #3

Address: A060 0A6C

D31:16	R	Default = 0x0000 0000	Reserved (read as 0)
D15:00	R/W	Default = 0x0000 0000	MFILTH3

Multicast High Address Filter Register #4

Address: A060 0A70

D31:16	R	Default = 0x0000 0000	Reserved (read as 0)
D15:00	R/W	Default = 0x0000 0000	MFILTH4

Multicast High Address Filter Register #5

Address: A060 0A74

D31:16	R	Default = 0x0000 0000	Reserved (read as 0)
D15:00	R/W	Default = 0x0000 0000	MFILTH5

Downloaded from www.Manualslib.com manuals search engine

Multicast High Address Filter Register #6

Address: A060 0A78

D31:16	R	Default = 0x0000 0000	Reserved (read as 0)
D15:00	R/W	Default = 0x0000 0000	MFILTH6

Multicast High Address Filter Register #7

Address: A060 0A7C

D31:16	R	Default = 0x0000 0000	Reserved (read as 0)
D15:00	R/W	Default = 0x0000 0000	MFILTH7

Multicast Address Mask registers

Each of the eight entries in the multicast address filter logic has individual mask registers that extend the filtering range of each entry. The multicast address mask for each entry is split between two registers. Each entry has a register that contains the lower 32 bits of the multicast mask and a separate register that contains the upper 16 bits of the mask.

- Bits are set to 1 in the mask to enable or include that bit in the address filter.
- Bits are set to 0 in the mask if they are not included or are disabled in the address filter. These bits become *don't cares.*

For an explanation of the synchronization scheme used for these registers, see "Clock synchronization" on page 276.

Multicast Low Address Mask Register #0

Address: A060 0A80

D31:00	R/W	Default = 0x0000 0000	MFMSKL0

Multicast Low Address Mask Register #1

Address: A060 0A84

D31:00	R/W	Default = 0x0000 0000	MFMSKL1

Multicast Low Address Mask Register #2

Address: A060 0A88

D31:00	R/W	Default = 0x0000 0000	MFMSKL2

Multicast Low Address Mask Register #3

Address: A060 0A8C

D31:00	R/W	Default = 0x0000 0000	MFMSKL3

Downloaded from www.Manualslib.com manuals search engine

Multicast Low Address Mask Register #4

Address: A060 0A90

D31:00	R/W	Default = 0x0000 0000	MFMSKL4

Multicast Low Address Mask Register #5

Address: A060 0A94

D31:00	R/W	Default = 0x0000 0000	MFMSKL5

Multicast Low Address Mask Register #6

Address: A060 0A98

D31:00	R/W	Default = 0x0000 0000	MFMSKL6

Multicast Low Address Mask Register #7

Address: A060 0A9C

D31:00	R/W	Default = 0x0000 0000	MFMSKL7

Multicast High Address Mask Register #0

Address: A060 0AA0

D31:16	R	Default = 0x0000 0000	Reserved (read as 0)
D15:00	R/W	Default = 0x0000 0000	MFMSKH0

Multicast High Address Mask Register #1

Address: A060 0AA4

D31:16	R	Default = 0x0000 0000	Reserved (read as 0)
D15:00	R/W	Default = 0x0000 0000	MFMSKH1

Multicast High Address Mask Register #2

Address: A060 0AA8

D31:16	R	Default = 0x0000 0000	Reserved (read as 0)
D15:00	R/W	Default = 0x0000 0000	MFMSKH2

Multicast High Address Mask Register #3

Address: A060 0AAC

D31:16	R	Default = 0x0000 0000	Reserved (read as 0)
D15:00	R/W	Default = 0x0000 0000	MFMSKH3

Multicast High Address Mask Register #4

Address: A060 0AB0

D31:16	R	Default = 0x0000 0000	Reserved (read as 0)

Downloaded from www.Manualslib.com manuals search engine

D15:00	R/W	Default = 0x0000 0000	MFMSKH4

Multicast High Address Mask Register #5

Address: A060 0AB4

D31:16	R	Default = 0x0000 0000	Reserved (read as 0)
D15:00	R/W	Default = 0x0000 0000	MFMSKH5

Multicast High Address Mask Register #6

Address: A060 0AB8

D31:16	R	Default = 0x0000 0000	Reserved (read as 0)
D15:00	R/W	Default = 0x0000 0000	MFMSKH6

Multicast High Address Mask Register #7

Address: A060 0ABC

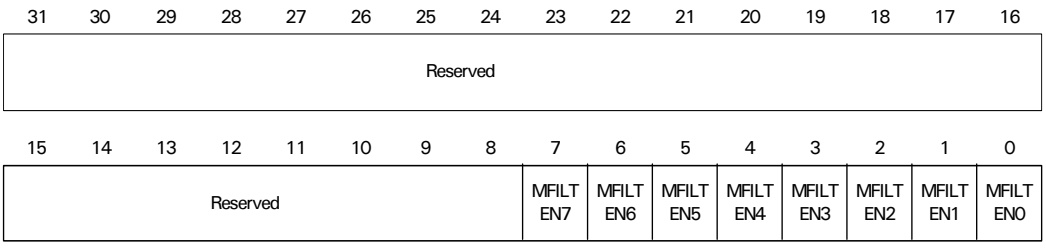
D31:16	R	Default = 0x0000 0000	Reserved (read as 0)
D15:00	R/W	Default = 0x0000 0000	MFMSKH7

Multicast Address Filter Enable register

Address: A060 0AC0

The Multicast Address Filter Enable register individually enables each of the eight entries in the multicast address filter logic. For an explanation of the synchronization scheme used for this register, see “Clock synchronization” on page 276.

Register


31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16
Reserved
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Reserved
MFILT EN7
MFILT EN6
MFILT EN5
MFILT EN4
MFILT EN3
MFILT EN2
MFILT EN1
MFILT EN0

Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:08	R	Reserved	N/A	**Read as 0**
D07	R/W	MFILTEN7	0x0000 0000	**Enable entry 7 of multicast address filter** 0 Disable entry 1 Enable entry
D06	R/W	MFILTEN6	0x0000 0000	**Enable entry 6 of multicast address filter** 0 Disable entry 1 Enable entry
D05	R/W	MFILTEN5	0x0000 0000	**Enable entry 5 of multicast address filter** 0 Disable entry 1 Enable entry
D04	R/W	MFILTEN4	0x0000 0000	**Enable entry 4 of multicast address filter** 0 Disable entry 1 Enable entry
D03	R/W	MFILTEN3	0x0000 0000	**Enable entry 3 of multicast address filter** 0 Disable entry 1 Enable entry
D02	R/W	MFILTEN2	0x0000 0000	**Enable entry 2 of multicast address filter** 0 Disable entry 1 Enable entry
D01	R/W	MFILTEN1	0x0000 0000	**Enable entry 1 of multicast address filter** 0 Disable entry 1 Enable entry
D00	R/W	MFILTEN0	0x0000 0000	**Enable entry 0 of multicast address filter** 0 Disable entry 1 Enable entry

TX Buffer Descriptor RAM

Address: A060 1000

The TX buffer descriptor RAM holds 64 transmit buffer descriptors on-chip. Each buffer descriptor occupies four locations in the RAM, and the RAM is implemented as a 256x32 device. This is the format of the TX buffer descriptor RAM:

Offset+0

D31:00	R/W	Source address

Downloaded from www.Manualslib.com manuals search engine

Offset+4

D31:11	R/W	Not used
D10:00	R/W	Buffer length

Offset+8

D31:00	R/W	Destination address (not used)

Offset+C

D31	R/W	W	Wrap
D30	R/W	I	Interrupt on buffer completion
D29	R/W	L	Last buffer on transmit frame
D28	R/W	F	Buffer full
D27:16	R/W	Reserved	N/A
D15:00	R/W	Status	Transmit status from MAC

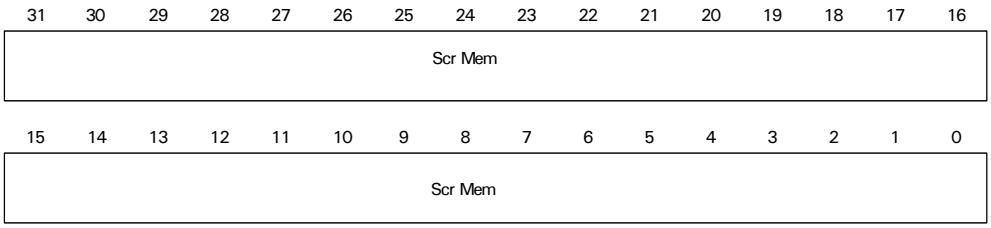
See “Transmit buffer descriptor format” on page 270, for more information about the fields in Offset+C.

RX FIFO RAM

Address: A060 2000 (512 locations)

The 2k Byte RX FIFO RAM can be used by the CPU as a scratch pad memory during boot up. CPU access is enabled by setting the RXRAM bit in the Ethernet General Control Register 1. This bit must be cleared before enabling the Ethernet receiver.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:00	R/W	Scr Mem	0	CPU scratch pad memory

Downloaded from www.Manualslib.com manuals search engine

Sample hash table code

This sample C code describes how to calculate hash table entries based on 6-byte Ethernet destination addresses and a hash table consisting of two 32-bit registers (HT1 and HT2). HT1 contains locations 31:0 of the hash table; HT2 contains locations 63:32 of the hash table.

The pointer to the hash table is bits [28:23] of the Ethernet destination address CRC. The polynomial is the same as that used for the Ethernet FCS:

G(x) = *x*^32+*x*^26+*x*^23+*x*^22+*x*^16+*x*^12+*x*^11+*x*^10+*x*^8+*x*^7+*x*^5+*x*^4+*x*^2+*x*+1

```
static ETH_ADDRESS mca_address[MAX_MCA];        /*list of MCA addresses*/
static INT16 mca_count;                         /*# of MCA addresses*/

/    *
     *
     * Function: void eth_load_mca_table (void)
     *
     * Description:
     *
     * This routine loads the MCA table. It generates a hash table for
     * the MCA addresses currently registered and then loads this table
     * into the registers HT1 and HT2.
     *
     * Parameters:
     *
     *       none
     *
     * Return Values:
     *
     *       none
     *
     */

static void eth_load_mca_table (void)

{
          WORD32 has_table[2];

          // create hash table for MAC address
          eth_make_hash_table (hash_table);
```


Downloaded from www.Manualslib.com manuals search engine

```
        (*MERCURY_EFE).ht2.bits.data = SWAP32(hash_table[1]);
        (*MERCURY_EFE).ht1.bits.data = SWAP32(hash_table[0]);
}


/   *
    *
    * Function: void eth_make_hash_table (WORD32 *hash_table)
    *
    * Description:
    *
    *       This routine creates a hash table based on the CRC values of
    *       the MAC addresses setup by set_hash_bit(). The CRC value of
    *       each MAC address is calculated and the lower six bits are used
    *       to generate a value between 0 and 64. The corresponding bit in
    *       the 64-bit hash table is then set.
    *
    *   Parameters:
    *
    *       hash_table              pointer to buffer to store hash table in.
    *
    * Return Values:
    *
    *       none
    *
    */

static void eth_make_hash_table (WORD32 *hash_table)

{
        int index;

        memset (hash_table, 0, 8);                      /* clear hash table*/

        for (index = 0; index < mca_count; index++)     /*for each mca address*/
        {
            set_hash_bit ((BYTE *) hash_table, calculate_hash_bit (mca_address
            [index]));
        }
}
```


Downloaded from www.Manualslib.com manuals search engine

```
/    *
     *
     * Function: void set_hash_bit (BYTE *table, int bit)
     *
     * Description:
     *
     *          This routine sets the appropriate bit in the hash table.
     *
     * Parameters:
     *
     *          table                   pointer to hash table
     *          bit                     position of bit to set
     *
     * Return Values:
     *
     *          none
     *
     */

static void set_hash_bit (BYTE *table, int bit)

{
          int byte_index, bit_index;

          byte-index = bit >> 3;
          bit_index = bit & 7;
          table [byte_index] |= (1 << bit_index);
}


/    *
     *
     * Function: int calculate_hash_bit (BYTE *mca)
     *
     * Description:
     *          This routine calculates which bit in the CRC hash table needs
     *          to be set for the MERCURY to recognize incoming packets with
     *          the MCA passed to us.
     *
     * Parameters:
     *
     *          mca                     pointer to multi-cast address
```


Downloaded from www.Manualslib.com manuals search engine

```
*
* Return Values:
*
*             bit position to set in hash table
*
*/

#define POLYNOMIAL 0x4c11db6L

static int calculate_hash_bit (BYTE *mca)

{
        WORD32 crc;
        WORD16 *mcap, bp, bx;
        int result, index, mca_word, bit_index;
        BYTE lsb;
        WORD16 copy_mca[3]
        memcpy (copy_mca,mca,sizeof(copy_mca));
        for (index = 0; index < 3; index++)
        {
            copy_mca [index] = SWAP16 (copy_mca [index]);
        }

        mcap = copy_mca;
        crc = 0xffffffffL;

        for (mca_word = 0; mca_word < 3; mca_word++)
        {
            bp = *mcap;
            mcap++;
            for (bit_index = 0; bit_index < 16; bit_index++)
            {
                    bx = (WORD16) (crc >> 16);          /* get high word of crc*/
                    bx = rotate (bx, LEFT, 1);          /* bit 31 to lsb*/
                    bx ^= bp;                           /* combine with incoming*/
                    crc <<= 1;                          /* shift crc left 1 bit*/
                    bx &= 1;                            /* get control bit*/
                    if (bx)                             /* if bit set*/
                    {
                     crc ^= POLYNOMIAL;                 /* xero crc with polynomial*/
                    }
                    crc |= bx:                          /* or in control bit*/
```


Downloaded from www.Manualslib.com manuals search engine

```
                    bp = rotate (bp, RIGHT, 1);
                }
            }

            // CRC calculation done. The 6-bit result resides in bit
            // locations 28:23

            result = (crc >> 23) & 0x3f;

            return result;

}
```


Downloaded from www.Manualslib.com manuals search engine

External DMA

CHAPTER 6

The external DMA interface provides two external channels for external peripheral support. Each DMA channel moves data from the source address to the destination address. These addresses can specify any peripheral on the AHB bus but, ideally, they specify an external peripheral and external memory.

DMA transfers

DMA transfers can be specified as burst-oriented to maximize AHB bus efficiency. All transfers are performed in two steps:

1. Data is moved from the source address to a 32-byte buffer in the DMA control logic.
2. The data is moved from the 32-byte buffer to the destination address.

These two steps are repeated until the DMA transfer is complete.

Note: Optimal performance is achieved when both the source address and destination address are aligned.

Initiating DMA transfers

DMA transfers can be initiated in one of two ways: processor-initiated and external peripheral initiated.

Processor-initiated

The processor must do these steps in the order shown:

1. Set up the required buffer descriptors.
2. Configure the DMA Control register for each channel.
3. Write a 1 to both the CE field and the CG field in the DMA Control register for each channel.

External peripheral-initiated

An external peripheral initiates a DMA transfer by asserting the appropriate REQ signal. Software must have set up the required buffer descriptors as well as the DMA Control register for each channel, including setting the CE field to 1, before the REQ signal can be asserted.

Downloaded from www.Manualslib.com manuals search engine

DMA buffer descriptor

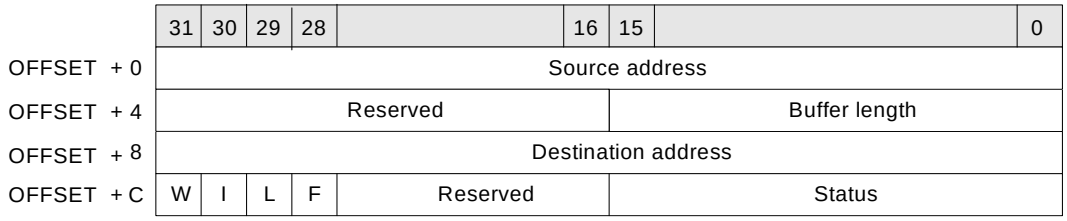
All DMA channels use a buffer descriptor. When a DMA channel is activated, it reads the DMA buffer descriptor that the Buffer Descriptor Pointer register points to. A DMA buffer descriptor is always fetched using an AHB INCR4 transaction to maximize AHB bus bandwidth. When the current descriptor is retired, the next descriptor is accessed from a circular buffer.

Each DMA buffer requires four 32-bit words to describe a transfer. Multiple buffer descriptors are located in circular buffers of 4096 bytes. The DMA channel's buffer descriptor pointer provides the first buffer descriptor address. Subsequent buffer descriptors are found adjacent to the first descriptor. The final buffer descriptor is defined with its W bit set. When the DMA channel finds the W bit, the channel wraps around to the first descriptor.

Each DMA channel can address a maximum of 256 buffer descriptors.

Important: A DMA channel configured for more than the maximum number of buffer descriptors operates in an unpredictable fashion.

DMA buffer descriptor diagram



Field descriptions follow.

Source address [pointer]

The source address pointer field identifies the starting location of the source data. The source address can be aligned to any byte boundary.

Note: Optimal performance is achieved when the source address is aligned on a word boundary.

Buffer length

Buffer length indicates the number of bytes to move between the source and the destination. After completing the transfer, the DMA controller updates this field with the actual number of bytes moved. This is useful for debugging error conditions or determining the number of bytes transferred before the DONE signal was asserted.

Destination address [pointer]

The description address pointer field identifies the starting location of the source data's destination; that is, to where the source data needs to be moved. The destination address can be aligned to any byte boundary.

Downloaded from www.Manualslib.com manuals search engine

Note: Optimal performance is achieved when the destination address is aligned on a word boundary.

Status This field is not used. Read back 0x0000.

Wrap (W) bit The Wrap (W) bit, when set, tells the DMA controller that this is the last buffer descriptor within the continuous list of descriptors. The next buffer descriptor is found using the initial DMA channel buffer descriptor pointer. When the W bit is not set, the next buffer descriptor is found using an offset of 0x10 from the current buffer descriptor.

Interrupt (I) bit The Interrupt (I) bit, when set, tells the DMA controller to issue an interrupt to the CPU when the buffer is closed due to a normal channel completion. The interruption occurs regardless of the normal completion interrupt enable configuration for the DMA channel.

Last (L) bit The Last (L) bit, when set, tells the DMA controller that this buffer descriptor is the last descriptor that completes an entire message frame. The DMA controller uses this bit to assert the normal channel completion status when the byte count reaches zero.

Full (F) bit The Full (F) bit, when set, indicates that the buffer descriptor is valid and can be processed by the DMA channel. The DMA channel clears this bit after completing the transfer(s).

The DMA channel does not try a transfer with the F bit clear. The DMA channel enters an idle state upon fetching a buffer descriptor with the F bit cleared. Whenever the F bit is modified by the device driver, the device driver must also write a 1 to the CE bit in the DMA Control register to activate the idle channel.

Descriptor list processing

Once a DMA controller has completed the operation specified by the current buffer descriptor, it clears the F bit and fetches the next buffer descriptor. A DMA channel asserts the NRIP field (buffer not ready interrupt pending) in the DMA Status register and returns to the idle state upon fetching a buffer descriptor with the F bit in the incorrect state. A DMA channel always closes the current descriptor and moves on to the next descriptor when a DMA transfer is terminated by the assertion of the DONE signal.

Downloaded from www.Manualslib.com manuals search engine

Peripheral DMA read access

The diagrams in this section describe how the DMA engine performs read accesses of an external peripheral.

- The CLK signal shown is for reference, and its frequency is equal to the speed grade of the part.
- The peripheral data enable signal (PDEN) is an AND function of the active states of the st_cs_n[n] and st_oe_n signals.
- PDEN timing can be adjusted by the memory controller's Static Memory Configuration 0-3 registers, which control st_cs_n[n] and st_oe_n.

Note: The PDEN signal is asserted for all accesses on the selected peripheral chip select. If configuration registers or memory also need to be accessed, you can use high level address bits and an external gate to disable the PDEN signal. You can also place the peripheral and configuration registers on separate chip selects to avoid the need for the external gate.

Determining the width of PDEN

DMA read accesses from an external peripheral are treated as asynchronous operations by the chip. It is critical that the necessary width of the PDEN assertion be computed correctly and programmed in the static memory controllers.

Use this equation to compute total access time:

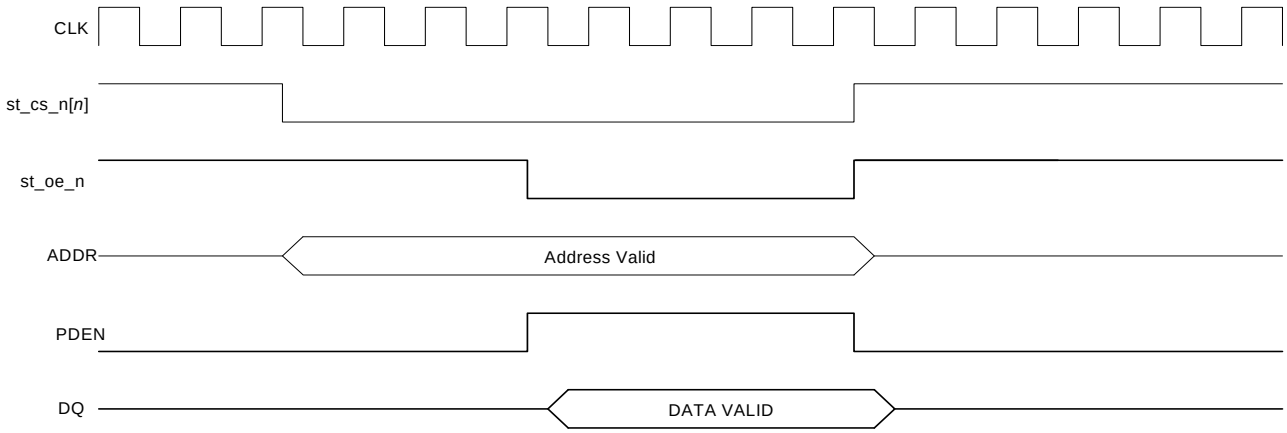
Total access time = $T_a + T_b + T_c + 10.0$

Equation variables

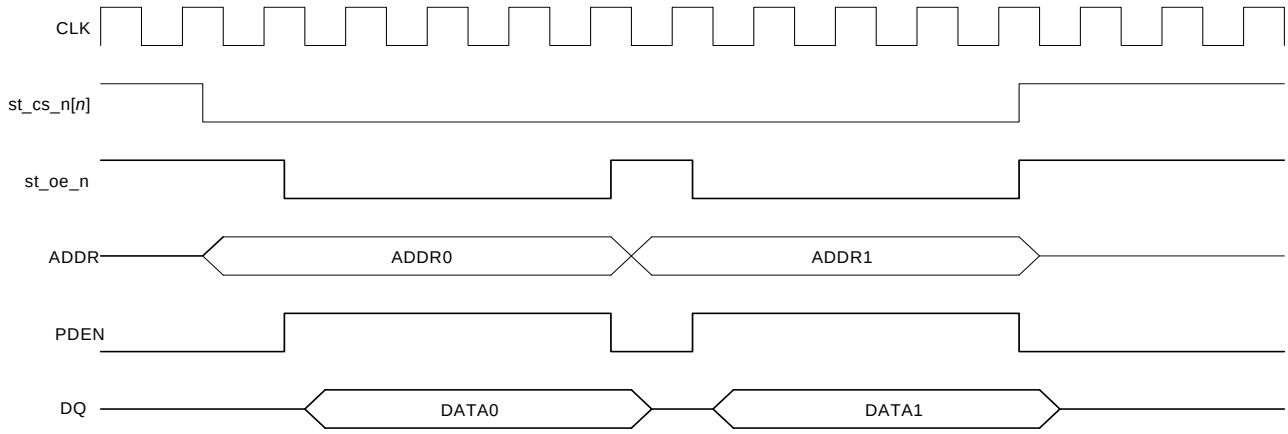
Variable	Definition
T_a	Peripheral read access time
T_b	Total board propagation delay including buffers
T_c	One AHB CLK cycle period

Downloaded from www.Manualslib.com manuals search engine

Peripheral DMA single read access



Peripheral DMA burst read access



Peripheral DMA write access

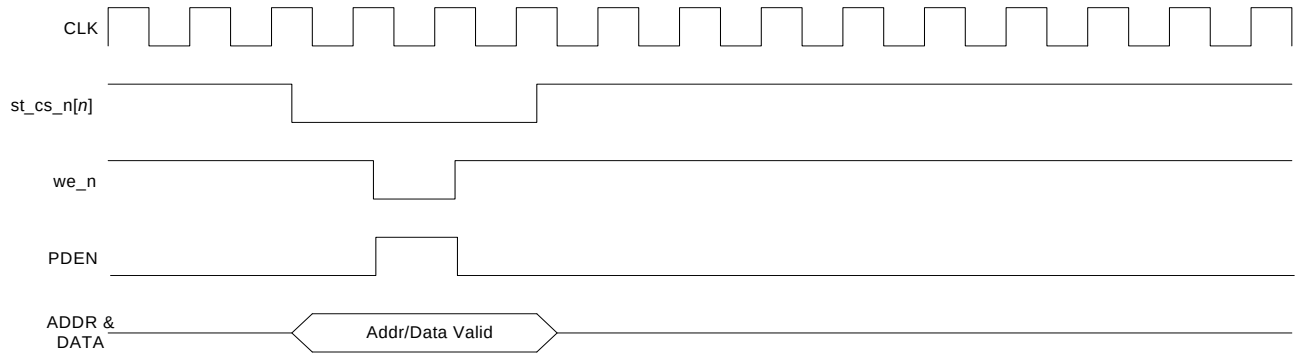
The diagrams in this section describe how the DMA engine performs write accesses of an external peripheral. The CLK signal shown is for reference, and its frequency is equal to the speed grade of the part. For peripheral writes, the PDEN signal is an AND function of the active status of st_cs_n[n] and we_n. Write data into the peripheral on the falling edge of the PDEN signal. Data and control signals are always held after the falling edge of PDEN for one reference CLK cycle.

Downloaded from www.Manualslib.com manuals search engine

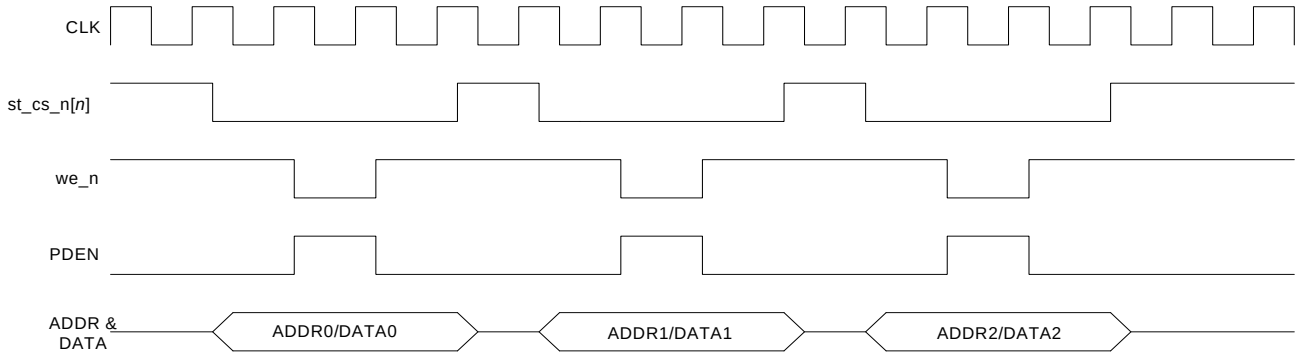
Determining the width of PDEN

Use the memory controller's Static Memory Write Delay register and Static Memory Write Enable Delay register to determine the width of the PDEN assertion.

Peripheral DMA single write access



Peripheral DMA burst write access



Peripheral REQ and DONE signaling

The processor treats the REQ and DONE signals as asynchronous, level signals.

REQ signal

- The external peripheral can initiate a DMA transfer at any time by asserting the REQ signal.
- The external peripheral can pause the DMA transfer at any time by deasserting the REQ signal.
- The REQ signal can be deasserted during a transfer but if the peripheral is configured for burst access, the burst completes. The DMA transfer control logic remains paused until the REQ signal is reasserted.

Downloaded from www.Manualslib.com manuals search engine

DONE signal

- The external peripheral can terminate the DMA transfer at any time by asserting the DONE signal. The peripheral must also deassert the REQ signal when it asserts the DONE signal.
- The DONE signal can be asserted during a transfer but if the peripheral is configured for burst access, the burst completes. When the DMA control logic finds a DONE assertion, it closes the current buffer descriptor, asserts a premature buffer completion status, and pauses until the REQ signal is reasserted. The DONE cycle must be deasserted no later that four AHB clock cycles before reasserting the REQ signal.

Special circumstances

- For memory-to-memory DMA transfers that are initiated by software writing a 1 to the channel go (CG) field in the DMA Control register, the DMA control logic ignores the REQ and DONE signals.
- For memory-to-peripheral transfers, the DMA control logic ignores the DONE signal.

Static RAM chip select configuration

The AHB DMA controller accesses an external peripheral using the external memory bus and one of the static ram chip select signals (st_cs_n[N]).

Static ram chip select configuration

This table shows how to program the static ram chip select control registers for access using the AHB DMA controller. Fields not explicitly listed must be left in the reset state. Fields listed but not defined must be defined by you.

Register name	Field	Value	Comment
Configuration	PB	1	System requirement
	PM	User-defined	Set to 1 if it is not necessary for the chip select signal to toggle for each access.
	MW	User-defined	
Read Delay	WTRD	User-defined	To determine the read delay: 1 Use this equation to compute the total delay: $T_a + T_b + T_c + 10.0$ 7 Divide the total delay by the AHB clock period 8 Round up any fractional value
Page Read Delay	WTPG	user-defined	For most applications, this is the same value as the WTRD value.

Downloaded from www.Manualslib.com manuals search engine

Register name	Field	Value	Comment
Output Enable Delay	WOEN	User-defined	For most applications, this field can be set to 0.
Write Enable Delay	WWEN	User-defined	For most applications, this field can be left in the default state.
Write Delay	WTWR	User-defined	For most applications, this field can be left in the default state.
Turn Delay	WTTN	User-defined	For most applications, this field can be left in the default state.

Control and Status registers

The external DMA configuration registers are located at base address 0xA080_0000. All the configuration registers are accessed with zero wait states.

Register address map

These are the external DMA control and status registers.

Address	Description	Access	Reset value
0xA080_0000	DMA Channel 1 Buffer Descriptor Pointer	R/W	0x00000000
0xA080_0004	DMA Channel 1 Control register	R/W	0x00000000
0xA080_0008	DMA Channel 1 Status and Interrupt Enable	R/W	0x00000000
0xA080_000C	DMA Channel 1 Peripheral Chip Select	R/W	0x00000000
0xA080_0010	DMA Channel 2 Buffer Descriptor Pointer	R/W	0x00000000
0xA080_0014	DMA Channel 2 Control register	R/W	0x00000000
0xA080_0018	DMA Channel 2 Status and Interrupt Enable	R/W	0x00000000
0xA080_001C	DMA Channel 2 Peripheral Chip Select	R/W	0x00000000

DMA Buffer Descriptor Pointer

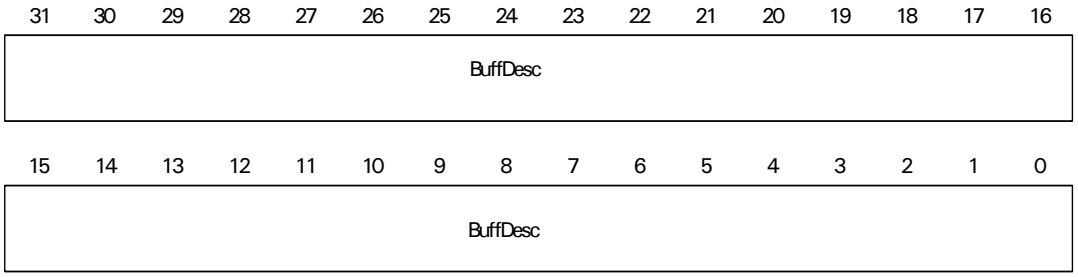
Address: A080_0000, A080_0010

The DMA Buffer Descriptor Pointer register contains a 32-bit pointer to the first buffer in a contiguous list of buffer descriptors.

The external DMA module has two of these registers. Each buffer descriptor is 16 bytes in length.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

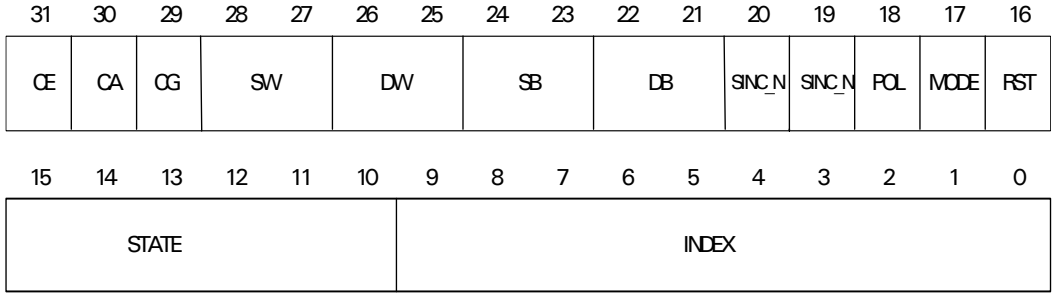
Bit(s)	Access	Mnemonic	Reset	Description
D31:00	R/W	BuffDesc	0x0000_0000	32-bit pointer to a buffer descriptor

DMA Control register

Address: A080_0004, A080_0014

The DMA Control register contains the required DMA transfer control information. The external DMA module has two of these registers.

Register



Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

Bit(s)	Access	Mnemonic	Reset	Description
D31	R/W	CE	0	**Channel enable** Enables and disables DMA operations as required. After a DMA channel has entered the IDLE state for any reason, this field must be written to a 1 to initiate further DMA transfers.
D30	R/W	CA	0	Channel abort When set, causes the current DMA operation to complete and closes the buffer.
D29	R/W	CG	0	**Channel go** When set, causes the DMA channel to exit the IDLE state and begin a DMA transfer. The CE field 31) must also be set, which allows software to initiate a memory-to-memory transfers. The dma_req and dma_done signals are not used during memory-to-memory transfers.
D28:27	R/W	SW	0	**Source width** Defines the data bus width of the device attached to the source address specified in the buffer descriptor. 00 8 bit 01 16 bit 10 32 bit 11 Reserved
D26:25	R/W	DW	0	**Destination width** Defines the data bus width of the device attached to the destination address specified in the buffer descriptor. 00 8 bit 01 16 bit 10 32 bit 11 Reserved
D24:23	R/W	SB	0	**Source burst** Defines the AHB maximum burst size allowed when reading from the source. Note that the source must have enough data, as defined by this register setting, before asserting REQ. 00 1 unit as set by the source width field (D28:27) 01 4 bytes (Recommended for 8-bit devices) 10 16 bytes (Recommended for 16-bit devices) 11 32 bytes (Recommended for 32-bit devices)

Downloaded from www.Manualslib.com manuals search engine

Bit(s)	Access	Mnemonic	Reset	Description
D22:21	R/W	DB	0	**Destination burst** Defines the AHB maximum burst size allowed when writing to the destination. Note that the destination must have enough space, as defined by this register setting, before asserting REQ. 00 1 unit as set by the destination width field (D26:25) 01 4 bytes (Recommended for 8-bit devices) 10 16 bytes (Recommended for 16-bit devices) 11 32 bytes (Recommended for 32-bit devices)
D20	R/W	SINC_N	0	**Source address increment** Controls whether the source address pointers are incremented after each DMA transfer. The DMA controller uses these bits in all modes whenever referring to a memory address. 0 Increment source address pointer 1 Do not increment source address pointer
D19	R/W	DINC_N	0	Destination address increment Controls whether the destination address pointers are incremented after each DMA transfer. The DMA controller uses these bits whenever referring to a memory address. 0 Increment destination address pointer 1 Do not increment destination address pointer
D18	R/W	POL	0	Control signal polarity Defines the active polarity of the dma_req, dma_done, and PDEN signals. 0 Active high signals 1 Active low signals
D17	R/W	MODE	0	**Fly-by mode** Defines the direction of data movement for fly-by DMA transfers. 0 Peripheral-to-memory fly-by-write DMA transfer 1 Memory-to-peripheral fly-by-read DMA transfer Note: This field is not used for DMA transfers initiated by writing a 1 to the CG field in this register (D29).

Downloaded from www.Manualslib.com manuals search engine

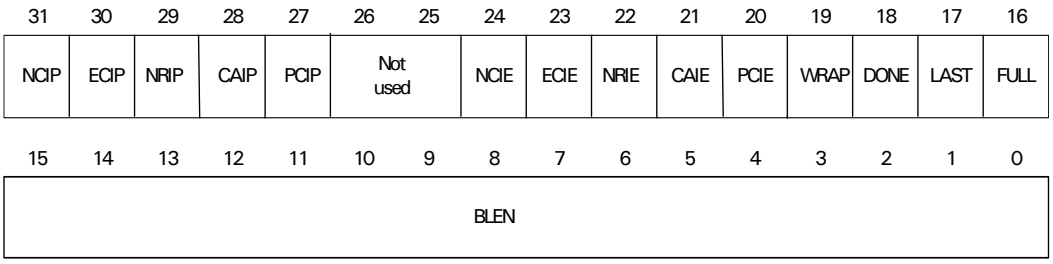
Bit(s)	Access	Mnemonic	Reset	Description
D16	R/W	RST	0	**Reset** Forces a reset of the DMA channel. Writing a 1 to this field forces all fields in this register, *except the index field*, to the reset state. The reset field is written with the value specified on signals HWDATA[9:0]. This field always reads back a 0. Note: Writing a 1 to this field while the DMA channel is operational will have unpredictable results.
D15:10	R	STATE	0	**State** 0 Idle 1-3 Buffer descriptor read 4-7 Data transfer 8-12 Buffer descriptor update 13 Error
D09:00	R/W	INDEX	0	**Index** Identifies the current 16-byte offset pointer relative to the buffer descriptor pointer. Note: This field can be written to only when the RST field (D16) is being written to a 1.

DMA Status and Interrupt Enable register

Address: A080_0008, A080_0018

The DMA Status and Interrupt Enable register contains the DMA transfer status and control information used in generating AHB DMA interrupt signals. The external DMA module has two of these registers.

Register



Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

Bit(s)	Access	Mnemonic	Reset	Description
D31	R/W1C	NCIP	0	**Normal completion interrupt pending** Set when a buffer descriptor has been closed. A normal DMA channel completion occurs when the BLEN count (D15:00) expires to zero and the L but in the buffer descriptor is set or when the peripheral device signals completion.
D30	R/W1C	ECIP	0	**Error completion interrupt pending** Set when the DMA channel encounters either a bad buffer descriptor pointer or a bad data buffer pointer. When the ECIP bit is set, the DMA channel stops until the ECIP bit is cleared by firmware. The DMA channel does not advance to the next buffer descriptor. When firmware clears the ECIP bit, the buffer descriptor is retried from where it left off. The CA bit in the DMA Control register can be used to abort the current buffer descriptor and advance to the next descriptor.
D29	R/W1C	NRIP	0	**Buffer not ready interrupt pending** Set when the DMA channel encounters a buffer descriptor whose F bit is in the incorrect state. The F bit must be set in order for the fetched buffer descriptor to be considered valid. If the F bit is not set, the descriptor is considered invalid and the NRIP field is set. When the NRIP bit is set, the DMA channel stops until the field is cleared by firmware. The DMA channel does not advance to the next buffer descriptor.
D28	R/W1C	CAIP	0	**Channel abort interrupt pending** Set when the DMA channel detects the CA bit (D30) set in the DMA Control register. When CAIP is set, the DMA channel stops until the CAIP bit is cleared by firmware. The DMA channel automatically advances to the next buffer descriptor after CAIP is cleared. The CA bit in the DMA Control register must be cleared, through firmware, before the CAIP bit is cleared. Failure to reset the CA bit cause the next buffer descriptor to abort also.
D27	R/W1C	PCIP	0	**Premature complete interrupt pending** Set when a DMA transfer is terminated by assertion of the dma_done signal. NCIP is set when PCIP is set for backwards compatibility.
D26:25	R/W	Not used	0	This field must always be set to 0.

Downloaded from www.Manualslib.com manuals search engine

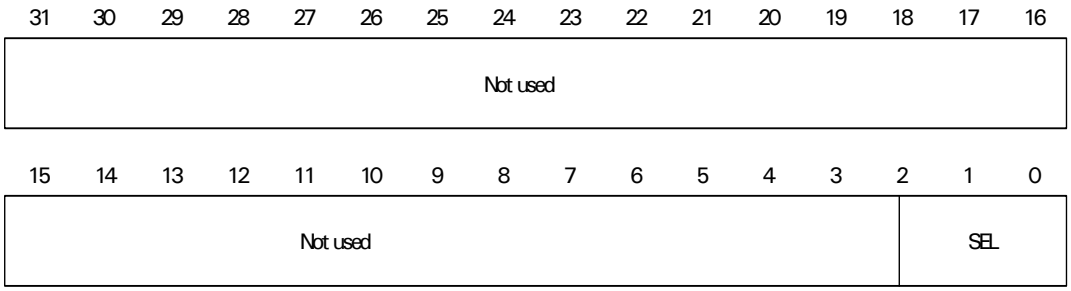
Bit(s)	Access	Mnemonic	Reset	Description
D24	R/W	NCIE	0	Enable NCIP interrupt generation.
D23	R/W	ECIE	0	Enable ECIE interrupt generation. This interrupt should always be enabled during normal operation.
D22	R/W	NRIE	0	Enable NRIP interrupt generation.
D21	R/W	CAIE	0	Enable CAIP interrupt generation. This interrupt should always be enabled during normal operation.
D20	R/W	PCIE	0	Enable PCIP interrupt generation.
D19	R	WRAP	0	*Read-only* debug field that indicates the last descriptor in the descriptor list.
D18	R	DONE	0	*Read-only* debug field that indicates the status of the DONE signal.
D17	R	LAST	0	*Read-only* debug field that indicates the last buffer descriptor in the current data frame.
D16	R	FULL	0	*Read-only* debug field that indicates the status of the F bit from the current DMA buffer descriptor.
D15:00	R	BLEN	0	*Read-only* debug field that indicates the current byte transfer count.

DMA Peripheral Chip Select register

Address: A080_000C, A080_001C

The DMA Peripheral Chip Select register contains the DMA channel peripheral chip select definition. The external DMA module has two of these registers.

Register



Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

Bit(s)	Access	Mnemonic	Reset	Definition
D31:02	R/W	Not used	0	This field must always be set to 0.
D01:00	R/W	SEL	0	**Chip select** Defines which of the four memory interface chip select signals (nmpmcstcsout[n]) is connected to the external peripheral. 00 nmpmcstcsout[0] 01 nmpmcstcsout[1] 10 nmpmcstcsout[2] 11 nmpmcstcsout[3]

Downloaded from www.Manualslib.com manuals search engine

Downloaded from www.Manualslib.com manuals search engine

AES Data Encryption/Decryption Module

CHAPTER 6

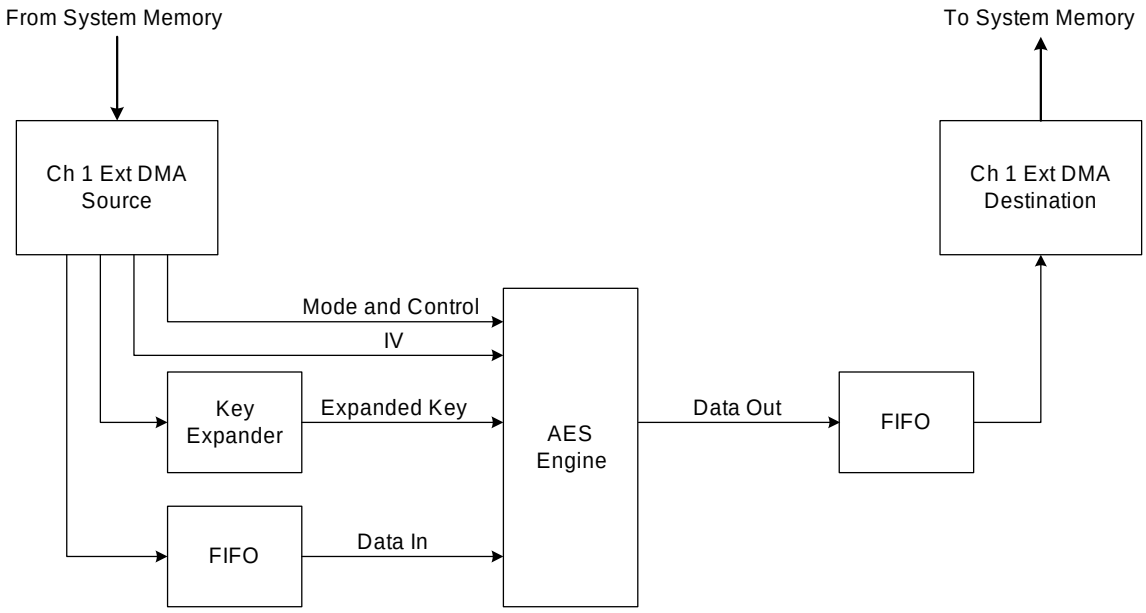
The AES data encryption/decryption module provides IPSec-compatible network security to processor-based systems. The AES core module implements Rijndael encoding/decoding in compliance with the NIST Advanced Encryption Standard (AES).

Features

- Processes 32 bits at a time.
- Is programmable for 128-, 192-, or 256-bit key lengths.
- Supports ECB, CBC, OFB, CTR, and CCM cipher modes.
- Implements a hardware key expander to minimize software intervention during the encryption/decryption process. During encryption and decryption, the key expander can produce the expanded key on the fly.
- Exists behind external DMA channel 1 (see Chapter 6, “External DMA,” for information about DMA control registers and programming).
- Uses the buffer descriptor control field to indicate a memory-to-memory AES operation.

Downloaded from www.Manualslib.com manuals search engine

Block diagram



Data blocks

The AES module works on 128-bit blocks of data. This table shows the performance per each 128-bit block, depending on the key size:

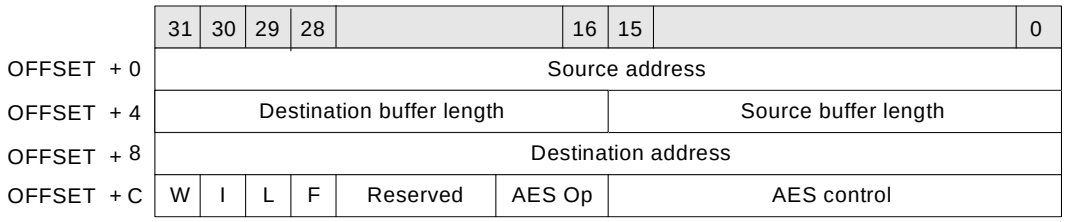
	Key size		
Characteristic	**128**	**192**	**256**
Number of cycles	44	52	60
Latency (cycles)	44	52	60
Throughput (bits/cycles)	~2.90	~2.46	~2.13
Throughput @ 75 MHz (bytes/sec)	~27.19	~23.06	~19.97

AES DMA buffer descriptor

The AES DMA buffer descriptor is the same as the external DMA buffer descriptor, with the exception of the control bits — AES op and AES control.

Downloaded from www.Manualslib.com manuals search engine

AES buffer descriptor diagram



Field definitions follow.

Source address [pointer]

The source address pointer identifies the starting location of the source data. The source address can be aligned to any byte boundary.

Note: Optimal performance is achieved when the source address is aligned on a word boundary.

Source buffer length

The source buffer length indicates the number of bytes to be read from the source. After completing the transfer, the DMA controller updates this field with the actual number of bytes that were moved. This is useful for debugging error conditions or determining the number of bytes transferred before the DONE signal was asserted.

Destination buffer length

The destination buffer length indicates the number of bytes to be written to the destination. This field should be identical to the source buffer length for all modes — with the exception of CCM — when the authentication code is being generated or a key is being expanded.

Destination address [pointer]

The description address pointer field identifies the starting location of the source data's destination; that is, to where the source data needs to be moved. The destination address must be word-aligned.

AES control

Bits	Used for	Values	
[2:0]	Encryption mode select	000	CBC
		001	CFB
		010	OFB
		011	CTR
		100	ECB
		101	CCM
		111	Key expand mode, which allows a key to be expanded by the hardware key expander and written back to system memory
[3]	Encryption/decryption select	0	Encryption
		1	Decryption

Downloaded from www.Manualslib.com manuals search engine

Bits	Used for	Values
[5:4]	Key size	00 128 bits 01 192 bits 10 256 bits
[6]	Additional authentication data (CCM mode only)	0 No additional data 1 Additional data used
[9:7]	L-par (CCM mode only)	N/A
[10]	Reserved	N/A
[13:11]	M-par (CCM mode only)	N/A
[15:14]	Reserved	N/A

AES op code

Indicates the contents of the data buffer associated with this descriptor:

- 000 Non-AES memory-to-memory or external DMA mode
- 001 Key buffer
- 010 IV buffer
- 011 Nonce buffer (CCM mode only, 16 bytes fixed length)
- 100 Additional authentication data (CCM mode only)
- 101 Data to be encrypted or decrypted

WRAP (W) bit

The Wrap (W) bit, when set, tells the DMA controller that this is the last buffer descriptor within the continuous list of descriptors. The next buffer descriptor is found using the initial DMA channel buffer descriptor pointer. When the W bit is not set, the next buffer descriptor is found using an offset of 0x10 from the current buffer descriptor.

Interrupt (I) bit

The Interrupt bit, when set, tells the DMA controller to issue an interrupt to the CPU when the buffer is closed due to a normal channel completion. The interrupt occurs regardless of the normal completion interrupt enable configuration for the DMA channel.

Last (L) bit

The Last bit, when set, tells the DMA controller that this buffer descriptor is the last descriptor that completes an entire message frame. The DMA controller uses this bit to assert the normal channel completion status when the byte count reaches zero.

Full (F) bit

The Full bit, when set, indicates that the buffer descriptor is valid and can be processed by the DMA channel. The DMA channel clears this bit after completing the transfer(s).

Downloaded from www.Manualslib.com manuals search engine

The DMA channel does not try a transfer when the F bit is clear. The DMA channel enters an idle state upon fetching a buffer descriptor with the F bit cleared.

When the F bit is modified by the device driver, the device driver must also write an 'I' to the CE bit (in the DMA Control register) to activate the idle channel.

Decryption

During decryption, the expanded key must be fed to the AES core backwards. The hardware key expander can handle this, but the input key is different than for encryption. The key must be expanded and the last words must be written to the key buffer as shown:

- A 128-bit key (K0, K1, K2, K3) is expanded to the following 32-bit word sequence: K0, K1, ..., K40, K41, K42, K43.

  To expand the key backwards, the hardware key expander needs K40-K43.

- A 192-bit key (K0, K1, K2, K3, K5, K6) is expanded to the following 32-bit word sequence: K0, K1, ..., K46, K47, K48, K49, K50, K51.

  To expand the key backwards, the hardware key expander core needs K48-51 followed by K46-47.

- A 256-bit key (K0, K1, K2, K3, K5, K6, K7) is expanded to the following 32-bit word sequence: K0, K1, ..., K52, K53, K54, K55, K56, K57, K58, K59.

  To expand the key backwards, the hardware key expander core needs K56-59 followed by K52-55.

The hardware key expander recreates all the remaining words in backwards order.

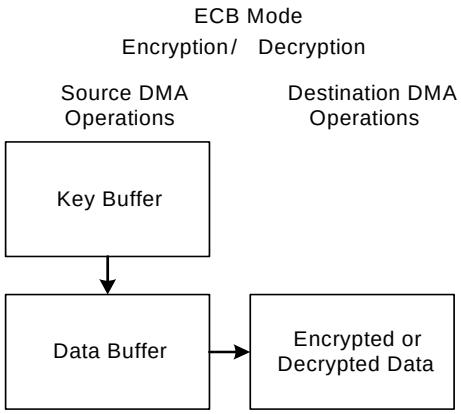
ECB processing

ECB mode does not require an initialization vector. Software just needs to set up a key buffer descriptor, followed by a data buffer descriptor.

Processing flow diagram

This is the ECB buffer descriptor processing flow:

Downloaded from www.Manualslib.com manuals search engine

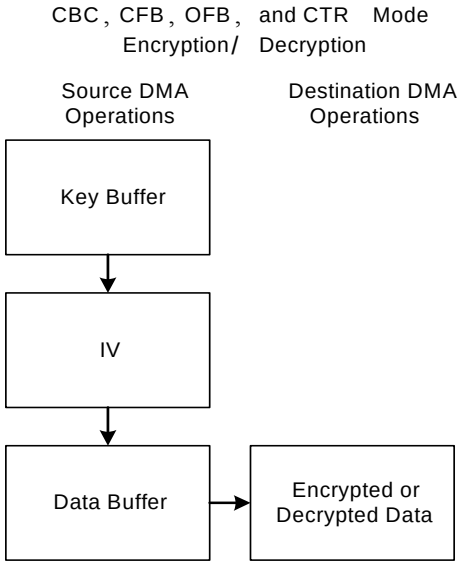


CBC, CFB, OFB, and CTR processing

CBC, CFB, OFB, and CTR modes need an initialization vector. Software must set up this buffer descriptor sequence: Key, IV, Data.

Processing flow diagram

This is the buffer descriptor processing flow for CBC, CFB, OFB, and CTR:



CCM mode

CCM mode does not require an initialization vector.

Downloaded from www.Manualslib.com manuals search engine

- For encryption, software must set up this buffer descriptor sequence: Key, Nonce, additional data (optional), data (used to compute the authentication code), data (used to perform the actual encryption).
- For decryption, software must set up this buffer descriptor sequence: Key, Nonce, Data (used to perform the actual decryption), Additional data (optional), Data (used to compute the authentication code).

Note: The data must be DMA'ed through the AES module twice in CCM mode for both encryption and decryption modes.

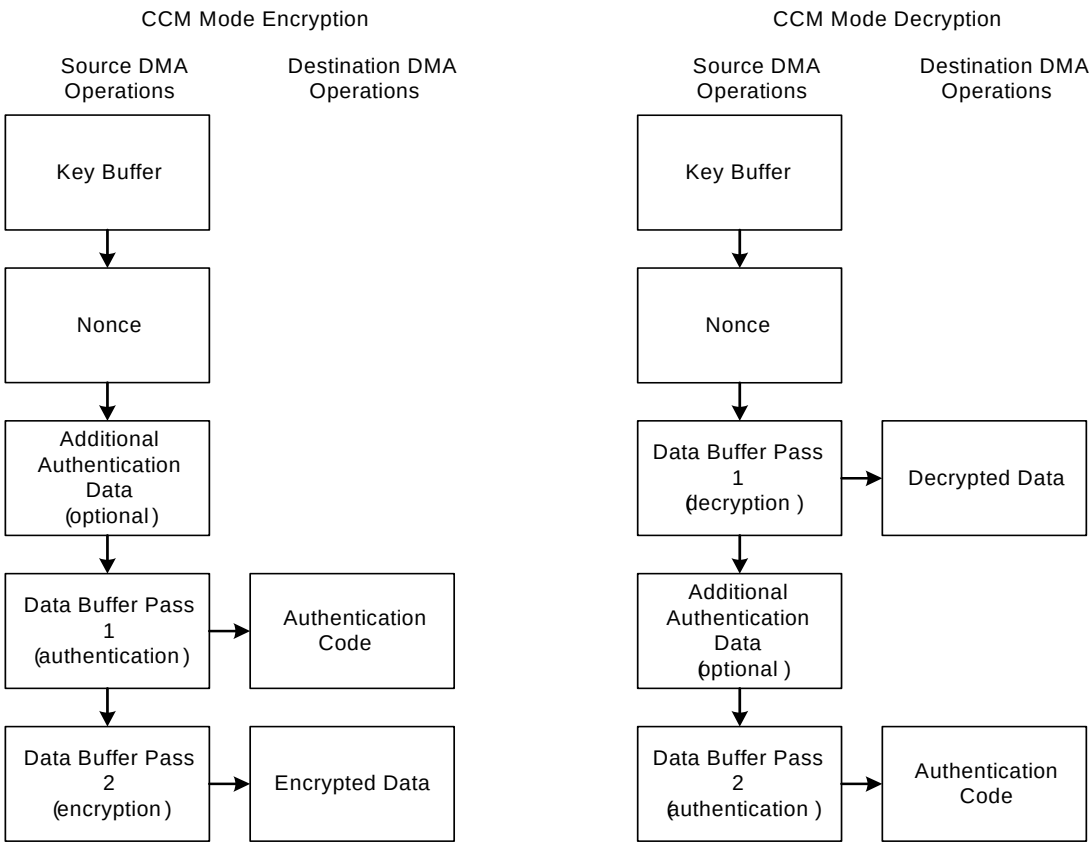
Nonce buffer

This is the format of the Nonce buffer:

Bits	127:120	119:8*L-par	8*L-par-1:0
Contents	reserved	Nonce	Message length

Processing flow

This is the CCM buffer descriptor processing flow:



Downloaded from www.Manualslib.com manuals search engine

Downloaded from www.Manualslib.com manuals search engine

I/O Hub Module

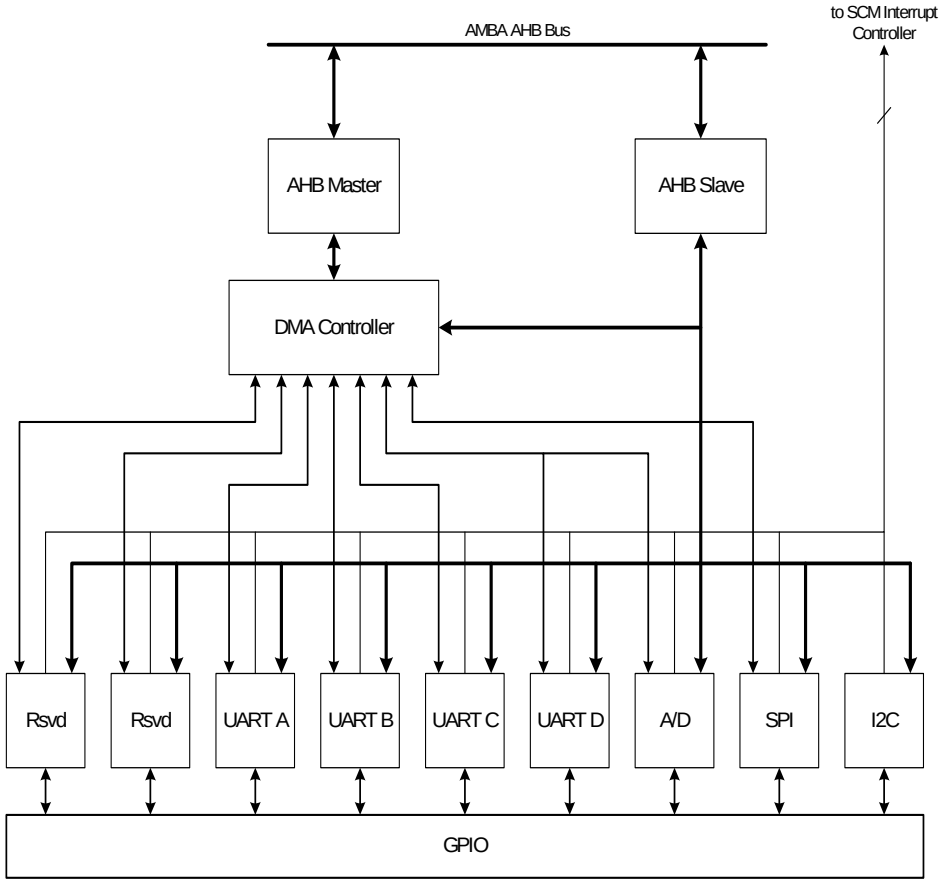
CHAPTER 9

The I/O hub provides access to the low speed ports on the processor through one master port on the AHB bus. The low speed ports include four UART ports, one SPI port, one I^2C port, 2 multi-function controlled ports, and one analog-to-digital (A/D) port. UART channel C can be configured for HDLC operation.

The SPI, UART, and A/D ports can be controlled either directly by the CPU or through the DMA controller, which is integrated into the I/O hub, The I^2C does not have DMA support.

Downloaded from www.Manualslib.com manuals search engine

Block diagram



AHB slave interface

The CPU has access to the control and status registers in the DMA controller, the peripheral devices, and the GPIO configuration.

DMA controller

The processor provides an eight channel DMA controller to service the low speed peripherals. Each channel has a transmit channel and a receive channel.

Servicing RX and FIFOs

The DMA controller services the RX and FIFOs in a round-robin manner. When one of the FIFOs needs servicing — that is, it can accept a burst of four 32-bit words — the DMA controller requests the AHB bus through the AHB master. After the request has been granted, the peripheral buffer data is transferred to or from system memory.

Downloaded from www.Manualslib.com manuals search engine

Buffer descriptors

The peripheral buffer data is held in buffers in external memory, linked together using buffer descriptors. The buffer descriptors are 16 bytes in length and are located contiguously in external memory.

This is the format of the buffer descriptor:

Address	Description	
offset + 0	Source address	
offset + 4	Reserved	Buffer length
offset + 8	Reserved	
offset + C	Control	Status

Source address [pointer]

The source address pointer points to the start of the buffer in system memory.

- For transmit channels, the address can start on any byte boundary.
- For receive channels, the address must be a 32-bit word aligned.

Buffer length

The buffer length is the length of the buffer in bytes, and allows a buffer size of up to 64k–1 bytes to be in a single buffer. Bits 31:16 are not used.

For receive channels, the buffer length field is updated with the actual number of bytes written to memory, as the peripheral has the ability to close the buffer early.

Control[15] – W

The Wrap (W) bit, when set, tells the DMA controller that this is the last buffer descriptor within the continuous list of descriptors. The next descriptor is found using the initial DMA channel buffer descriptor pointer. When the W bit is not set, the next buffer descriptor is found using the 16-byte offset.

Control[14] – I

The Interrupt (I) bit, when set, tells the DMA controller to issue an interrupt when the buffer is closed due to normal channel completion.

Control[13] – L

This is the Last (L) bit.

- For transmit channels, firmware sets the L bit when the current buffer is the last in the packet.
- For receive channels. hardware sets the L bit when the current buffer is closed by status bits received from the peripheral device. Status bits can include conditions such as a character gap timeout, character match, or error condition.

Control[12] – F

This is the Full (F) bit.

Downloaded from www.Manualslib.com manuals search engine

- For transmit channels. CPU sets the F bit after the data is written to a buffer. The DMA controller clears this bit as each buffer is read from external memory. If the DMA controller ever finds that this bit is not set when the buffer descriptor is read, the NRIP bit is set in the Interrupt Status register and the DMA controller stops immediately and goes to the ERROR state. The CPU must clear the CE bit to restore the DMA.
- For receive channels, hardware sets the F bit after data is written to a buffer. The CPU must clear the F bit after all data has been read from the buffer. If the DMA controller ever finds that this bit is not clear when the buffer descriptor is read, the NRIP bit is set in the Interrupt Status register and the DMA controller stops immediately.

  The DMA controller must be soft reset after the buffer descriptor problem has been solved.

Control[11:0] These bits are not used.

Status[15:0] The status depends on the module, as defined in the next tables.

Note: In direct mode, the status can be read from the Direct Mode RX Status FIFO.

UART

Bits	Description
15:7	Reserved
6:5	01 Error; bits 3:0 indicate the error type bit 4: Reserved bit 3: Receiver overflow, should never occur in a properly configured system bit 2: Parity error bit 1: Framing error bit 0: Break condition

Downloaded from www.Manualslib.com manuals search engine

HDLC

Bits	Description
15:7	Reserved
6:5	01 HDLC frame close, bits 3:0 indicate the close condition bit 4: The last byte is less than 8 bits bit 3: Receiver overflow, should never occur in a properly configured system bit 2: Invalid CRC found at end of frame bit 1: Valid CRC found at end of frame bit 0: Abort condition found 11 match character found bit 4: Match character 4 bit 3: Match character 3 bit 2: Match character 2 bit 1: Match character 1 bit 0: Match character 0 00 Other close event bit 2: Buffer gap timer expired bit 1: Software-initiated buffer close

SPI

Not applicable.

Transmit DMA example

After the last buffer in the data packet has been placed in system memory and the buffer descriptors have been configured, the data packet is ready to be transmitted. The CPU configures the module DMA TX buffer descriptor pointer, TXBDP (see "[Module] DMA TX Buffer Descriptor Pointer" on page 381), and then sets the channel enable bit in the DMA Control register.

Process

The DMA controller starts the process to read the buffer descriptor and buffer data from system memory using the AHB master. The DMA controller follows this process:

1 Reads the first buffer descriptor, as pointed to by the TX buffer descriptor pointer and INDEX.

Downloaded from www.Manualslib.com manuals search engine

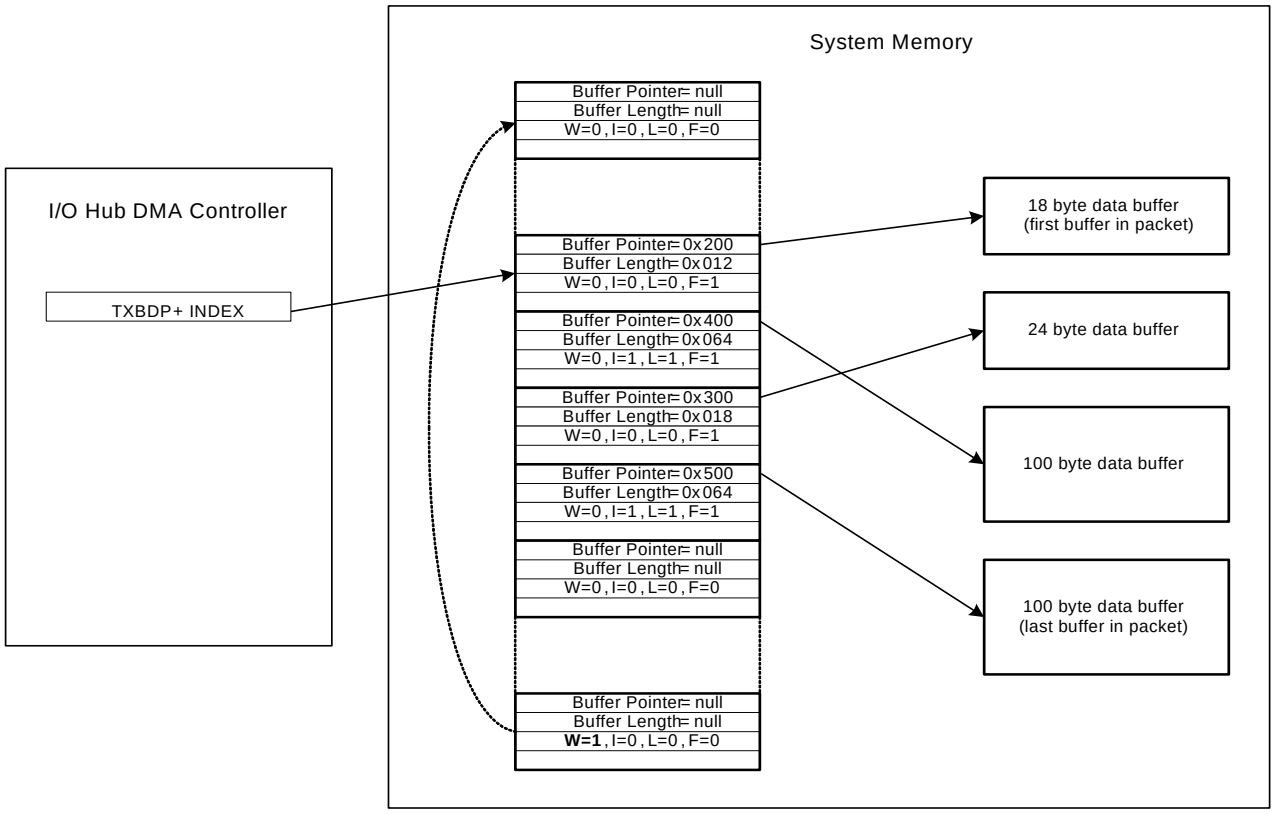
2 Verifies that the data buffer is valid by making sure the F bit is set to 1.

3 Reads the first data buffer, in 16-byte bursts.

4 Continues to process the buffer descriptors and data buffers until all data has been transmitted from the buffer descriptor with the L bit set to 1. The DMA controller interrupts the CPU if the I bit is set to a 1.

5 Remains in the IDLE state until the channel enable bit is set to a 0, then set to a 1 again.

Visual example



Control and status register address maps

The I/O Hub provides a series of registers for the low speed peripheral modules it supports. The DMA, direct mode, and interrupt control register formats are the same for these modules. The base address for the registers is 0x9000_0000. Write buffering in the MMU must be disabled for all registers in the I/O Hub address space, from address 0x9000_0000 to 0x9FFF_FFFF.

Register address maps are shown for each low speed peripheral module.

Downloaded from www.Manualslib.com manuals search engine

Note: Registers 9000_0000 – 9000_7FFF and registers 9000_8000 – 9000_FFFF are reserved.

UART A register address map

Register Offset	Description (31:00)
0x9001_0000	UART A Interrupt and FIFO Status
0x9001_0004	UART A DMA RX Control
0x9001_0008	UART A DMA RX Buffer Descriptor Pointer
0x9001_000C	UART A DMA RX Interrupt Configuration register
0x9001_0010	UART A Direct Mode RX Status FIFO
0x9001_0014	UART A Direct Mode RX Data DIDO
0x9001_0018	UART A DMA TX Control
0x9001_001C	UART A DMA TX Buffer Descriptor Pointer
0x9001_0020	UART A DMA TX Interrupt Configuration register
0x9001_0024	Reserved
0x9001_0028	UART A Direct Mode TX Data FIFO
0x9001_002C	UART A Direct Mode TX Data Last FIFO
0x9001_0030 – 0x9001_0FFF	Reserved
0x9001_1000 – 0x9001_7FFF	UART A CSR Space

UART B register address map

Register Offset	Description (31:00)
0x9001_8000	UART B Interrupt and FIFO Status
0x9001_8804	UART B DMA RX Control
0x9001_8008	UART B DMA RX Buffer Descriptor Pointer
0x9001_800C	UART B DMA RX Interrupt Configuration register
0x9001_8010	UART B Direct Mode RX Status FIFO
0x9001_8014	UART B Direct Mode RX Data FIFO
0x9001_8018	UART B DMA TX Control
0x9001_801C	UART B DMA TX Buffer Descriptor Pointer
0x9001_8020	UART B DMA TX Interrupt Configuration register
0x9001_8024	Reserved
0x9001_8028	UART B Direct Mode TX Data FIFO
0x9001_802C	UART B Direct Mode TX Data Last FIFO

Downloaded from www.Manualslib.com manuals search engine

Register Offset	Description (31:00)
0x9001_8030 – 0x9001_8FFF	Reserved
0x9001_9000 – 0x9001_9FFF	UART B CSR Space

UART C register address map

Register Offset	Description (31:00)
0x9002_0000	UART C Interrupt and FIFO Status
0x9002_0004	UART C DMA RX Control
0x9002_0008	UART C DMA RX Buffer Descriptor Pointer
0x9002_000C	UART C DMA RX Interrupt Configuration register
0x9002_0010	UART C Direct Mode RX Status FIFO
0x9002_0014	UART C Direct Mode RX Data FIFO
0x9002_0018	UART C DMA TX Control
0x9002_001C	UART C DMA TX Buffer Descriptor Pointer
0x9002_0020	UART C DMA TX Interrupt Configuration register
0x9002_0024	Reserved
0x9002_0028	UART C Direct Mode TX Data FIFO
0x9002_002C	UAT C Direct Mode TX Data Last FIFO
0x9002_0030 – 0x9002_0FFF	Reserved
0x9002_1000 – 0x9002_7FFF	UART C CSR Space

UART D register address map

Register Offset	Description (31:00)
0x9002_8000	UART D Interrupt and FIFO Status
0x9002_8004	UART D DMA RX Control
0x9002_8008	UART D DMA RX Buffer Descriptor Pointer
0x9002_800C	UART D DMA RX Interrupt Configuration register
0x9002_8010	UART D DIrect Mode RX Status FIFO
0x9002_8014	UART D Direct Mode RX Data FIFO
0x9002_8018	UART D DMA TX Control
0x9002_801C	UART D DMA TX Buffer Descriptor Pointer
0x9002_8020	UART D DMA TX Interrupt Configuration register
0x9002_8024	Reserved
0x9002_8028	UART D Direct Mode TX Data FIFO
0x9002_802C	UART D Direct Mode TX Data Last FIFO

Downloaded from www.Manualslib.com manuals search engine

Register Offset	Description (31:00)
0x9002_8030 – 0x9002_8FFF	Reserved
0x9002_9000 – 0x9002_FFFF	UART D CSR Space

SPI register address map

Register Offset	Description (31:00)
0x9003_0000	SPI Interrupt and FIFO Status
0x9003_0004	SPI DMA RX Control
0x9003_0008	SPI DMA RX Buffer Descriptor Pointer
0x9003_000C	SPI DMA RX Interrupt Configuration register
0x9003_0010	SPI Direct Mode RX Status FIFO
0x9003_0014	SPI Direct Mode RX Data FIFO
0x9003_0018	SPI DMA TX Control
0x9003_001C	SPI DMA TX Buffer Descriptor Pointer
0x9003_0020	SPI DMA TX Interrupt Configuration register
0x9003_0024	Reserved
0x9003_0028	SPI Direct Mode TX Data FIFO
0x9003_002C	SPI Direct Mode TX Data Last FIFO
0x9003_0030 – 0x9003_0FFF	Reserved
0x9003_1000 – 0x9003_7FFF	SPI CSR Space

AD register address map

Register Offset	Description (31:00)
0x9003_8000 – 0x9003_8FFF	Reserved
0x9003_9000 – 0x9003_FFFF	AD CSR Space

Reserved

Registers 9004_0000 – 9004_7FFF and 9004_8000 – 9004_FFFF are reserved.

I^2C register address map

Register Offset	Description (31:00)
0x9005_0000 – 0x9005_7FFF	I^2C CSR Space

Reserved

Registers 9005_8000 – 9005_FFFF are reserved.

Downloaded from www.Manualslib.com manuals search engine

RTC register address map

Register Offset	Description (31:00)
0x9006_0000 – 0x9006_00BF	RTC CSR Space
0x9006_0000 – 0x9006_00FC	64-byte Battery Backed RAM

IO Hardware Assist register address map (0)

Register Offset	Description (31:00)
0x9006_8000 – 0x9006_FFFF	IO Hardware Assist CSR Space for Flexible I/O Module 0

IO Hardware Assist register address map (1)

Register Offset	Description (31:00)
0x9007_0000 – 0x9007_7FFF	IO Hardware Assist CSR Space for Flexible I/O Module 1

IO register address map (0)

Register Offset	Description (31:00)
0x9008_0000 – 0x9008_FFFF	IO Space for Flexible I/O Module 0

IO register address map (1)

Register Offset	Description (31:00)
0x9009_0000 – 0x9009_FFFF	IO Space for Flexible I/O Module 1

[*Module*] Interrupt and FIFO Status register

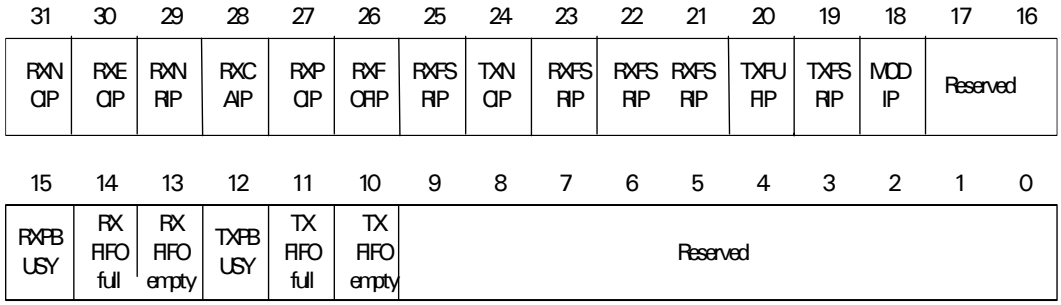
Addresses: 9000_0000 / 9000_8000 / 9001_0000 / 9001_8000 / 9002_0000 / 9002_8000 / 9003_0000 / 9003_8000

The Interrupt and FIFO Status register allows software to determine the cause of the current low speed peripheral interrupts and to clear the interrupt bit.

Note: An access type of R/W* means that the processor must write 1 to clear the value if the read value is 1. If the read value is 0, the write value must be 0.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

Bit(s)	Access	Mnemonic	Reset	Description
D31	R/W*	RXNCIP	0x0	**Normal completion interrupt pending (RX)** Set when a buffer is closed under normal conditions. An interrupt is generated when the I bit is set in the current buffer descriptor. A normal DMA completion occurs when the buffer length field expires.
D30	R/W*	RXECIP	0x0	**Error completion interrupt pending (RX)** Set when the DMA channel finds either a bad buffer descriptor or a bad data buffer pointer. The DMA channel remains in the ERROR state until the CE bit in the DMA Control register is cleared and then set again. The DMA channel then uses the buffer descriptor as set in the index control field.
D29	R/W*	RXNRIP	0x0	**Buffer not ready interrupt pending (RX)** Set when he DMA channel finds a buffer descriptor with the F bit not set. The DMA channel remains in the ERROR state until the CE bit is set in the DMA Control register is cleared and then set again.
D28	R/W*	RXCAIP	0x0	**Channel abort interrupt pending (RX)** Set when the DMA channel finds the channel abort (CA) bit set. The DMA controller closes the current buffer descriptor and remains in the IDLS state until the CA bit is cleared and the CE bit is set.
D27	R/W*	RXPCIP	0x0	**Premature completion interrupt pending** Set when a buffer descriptor is closed by the peripheral instead of by reaching the buffer length. The DMA channel continues processing buffer descriptors

Downloaded from www.Manualslib.com manuals search engine

Bit(s)	Access	Mnemonic	Reset	Description
D26	R/W*	RXFOFIP	0x0	**RX FIFO overflow interrupt pending** Set when the RX FIFO finds an overflow condition.
D25	R/W*	RXFSRIP	0x0	**RX FIFO service request interrupt pending (RX)** Set when the RX FIFO level rises above the receive FIFO threshold (in the RX Interrupt Configuration register).
D24	R/W*	TXNCIP	0x0	**Normal completion interrupt pending (TX)** Set when a buffer is closed under normal conditions. An interrupt is generated when the I bit is set in the current buffer descriptor. A normal DMA completion occurs when the buffer length field expires.
D23	R/W*	TXECIP	0x0	**Error completion interrupt pending (TX)** Set when the DMA channel finds either a bad buffer descriptor or a bad data buffer pointer. The DMA channel remains in the ERROR state until the CE bit in the DMA Control register is cleared and then set again. The DMA channel then uses the buffer descriptor as set in the index control field.
D22	R/W*	TXNRIP	0x0	**Buffer not ready interrupt pending (TX)** Set when the DMA channel finds a buffer descriptor with the F bit not set. The DMA channel remains in the ERROR state until the CE bit in the DMA Control register is cleared and then set again. The DMA channel then uses the buffer descriptor as set in the index control field.
D21	R/W*	TXCAIP	0x0	**Channel abort interrupt pending (TX)** Set when the DMA channel finds the channel abort (CA) control bit set. The DMA controller closes the current buffer descriptor and remains in the IDLE state until the CA bit is cleared and the CE bit is set.
D20	R/W*	TXFUFIP	0x0	**TX FIFO underflow interrupt pending** Set when the TX FIFO finds an underflow.
D19	R/W*	TXFSRIP	0x0	**TX FIFO service request interrupt pending (TX)** Set when the TX FIFO level drops below the transmit FIFO threshold (in the TX Interrupt Configuration register).

Downloaded from www.Manualslib.com manuals search engine

Bit(s)	Access	Mnemonic	Reset	Description
D18	R	MODIP	0x0	**Module interrupt pending** The hardware module has asserted an interrupt. Software must read the appropriate Interrupt Status register to determine the cause.
D17:16	N/A	Reserved	N/A	N/A
D15	R	RXPBUSY	0x0	0 Peripheral idle 1 Peripheral busy Note: Applicable only for channels connected to the flexible I/O module processors. The CPU must not access the Module Direct Mode RX Data FIFO Read register when this bit is set. If this bit is set, the read generates a bus error.
D14	R	RX FIFO full	0x0	**Receive status and data FIFO full status** 0 Not full 1 Full
D13	R	RX FIFO empty	0x1	**Receive status and data FIFO empty status** 0 Not empty 1 Empty
D12	R	TXPBUSY	0x0	0 Peripheral idle 1 Peripheral busy Note: Applicable only for channels connected to the flexible I/O module processors. The CPU must not access the Module Direct Mode TX Data FIFO register when this bit is set. If this bit is set, the read generates a bus error.
D11	R	TX FIFO full	0x0	**Transmit data FIFO full status** 0 Not full 1 Full
D10	R	TX FIFO empty	0x1	**Transmit data FIFO empty status** 0 Not empty 1 Empty
D09:00	N/A	Reserved	N/A	N/A

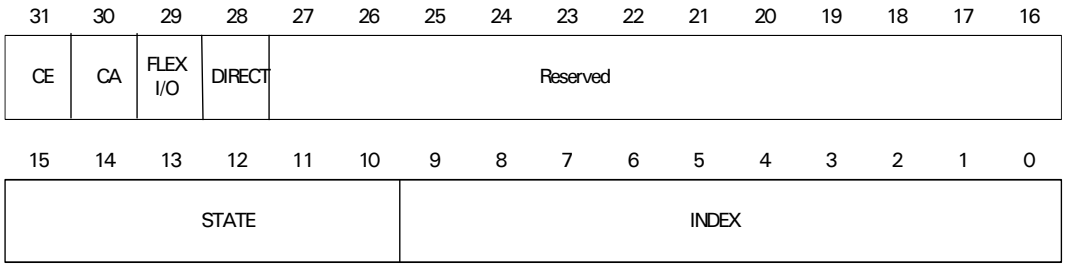
[*Module*] DMA RX Control

Addresses: 9000_0004 / 9000_8004 / 9001_0004 / 9001_8004 / 9002_0004 / 9002_8004 / 9003_0004 / 9003_8004

The DMA RX Control register contains control register settings for each receive DMA channel.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

Bit(s)	Access	Mnemonic	Reset	Description
D31	R/W	CE	0x0	**Channel enable** 0 Disable DMA operation 1 Enable DMA operation
D30	R/W	CA	0x0	**Channel abort** When set, causes the current DMA operation to complete and closes the buffer. The DMA channel remains idle until this bit is cleared.
D29	R/W	FLEX I/O	0x0	0 DMA controlled by CPU 1 DMA controlled by flexible I/O This bit is valid only for channels 0 and 1, which are assigned to flexible I/O module 0 and flexible I/O module 1.
D28	R/W	DIRECT	0x0	0 DMA mode 1 Direct access mode
D27:16	N/A	Reserved	N/A	N/A
D15:10	R	STATE	0x0	DMA state machine status field
D09:00	R	INDEX	0x0	This field can be read at any time to determine the current index.

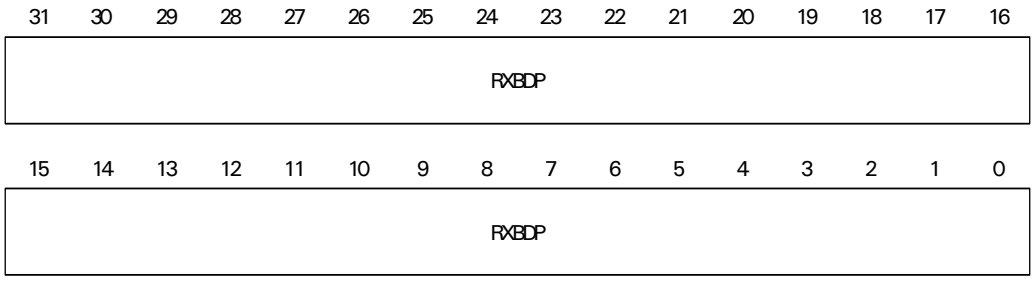
[*Module*] DMA RX Buffer Descriptor Pointer

Addresses: 9000_0008 / 9000_8008 / 9001_0008 / 9001_8008 / 9002_0008 / 9002_8008 / 9003_0008 / 9003_8008

The DMA RX Buffer Descriptor Pointer register is the address of the first buffer descriptor for each DMA channel.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

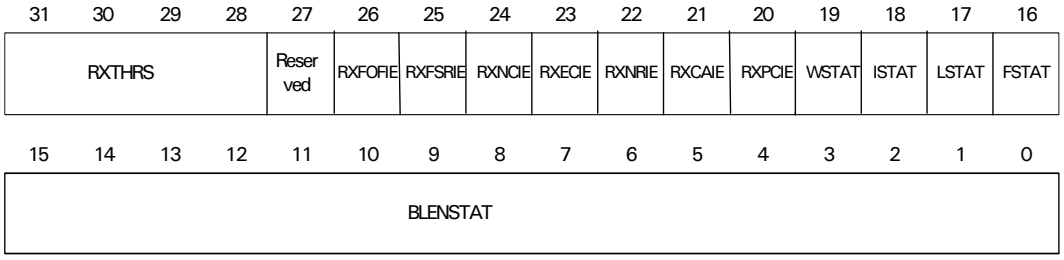
Bit(s)	Access	Mnemonic	Reset	Description
D31:00	R/W	RXBDP	0x0	The first buffer descriptor in the ring. Used when the W bit is found, which indicates the last buffer descriptor in the list.

[*Module*] RX Interrupt Configuration register

Addresses: 9000_000C / 9000_800C / 9001_000C / 9001_800C / 9002_000C / 9002_800C / 9003_000C / 9000_800C

The RX Interrupt Configuration register allows system software to configure the interrupt for the I/O hub module receive channel.

Register



Register bit assignment

Bit(s)	Access	Mnemonic	Reset	Description
D31:28	R/W	RXTHRS	0xF	**RX FIFO threshold** An interrupt is generated when the FIFO level rises above this level.
D27	N/A	Reserved	N/A	N/A

Downloaded from www.Manualslib.com manuals search engine

Bit(s)	Access	Mnemonic	Reset	Description
D26	R/W	RXFOFIE	0x0	Enable the RXFOFIP interrupt.
D25	R/W	RXFSRIE	0x0	Enable the RXFSRIP interrupt.
D24	R/W	RXNCIE	0x0	Enable the RXNCIP interrupt.
D23	R/W	RXECIE	0x0	Enable the RXECIP interrupt.
D22	R/W	RXNRIE	0x0	Enable the RXNRIP interrupt.
D21	R/W	RXCAIE	0x0	Enable the RXCAIP interrupt.
D20	R/W	RXPCIE	0x0	Enable the RXPCIP interrupt.
D19	R	WSTAT	0x0	Debug field, indicating the W bit is set in the current buffer descriptor.
D18	R	ISTAT	0x0	Debug field, indicating the I bit is set in the current buffer descriptor.
D17	R	LSTAT	0x0	Debug field, indicating the L bit is set in the current buffer descriptor.
D16	R	FSTAT	0x0	Debug field, indicating the F bit is set in the current buffer descriptor.
D15:00	R	BLENSTAT	0x0	Debug field, indicating the current byte count.

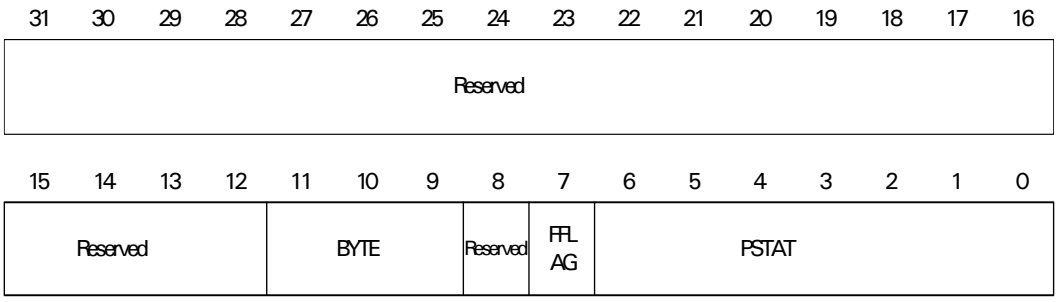
[*Module*] Direct Mode RX Status FIFO

Addresses: 9000_0010 / 9000_8010 / 9001_0010 / 9001_8010 / 9002_0010 / 9002_8010 / 9003_0010 / 9003_8010

The Direct Mode RX Status FIFO register is used when in direct mode of operation, to determine the status of the receive FIFO.

- This register must be read before each read to the RX Data FIFO register.
- The RX Data FIFO register must be read after each read to this register, even if the BYTE field is 0.

Register



Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

Bit(s)	Access	Mnemonic	Reset	Description
D31:12	N/A	Reserved	N/A	N/A
D11:09	R	BYTE	N/A	Number of bytes in the current 32-bit location.
D08	N/A	Reserved	N/A	N/A
D07	R	FFLAG	N/A	**Full flag** Indicates that the FIFO went full when the current location was written.
D06:00	R	PSTAT	N/A	General peripheral status, unique to the peripheral attached to the channel.

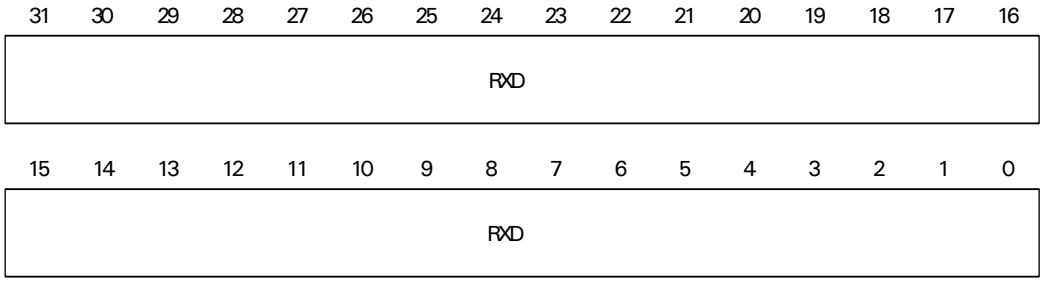
[*Module*] Direct Mode RX Data FIFO

Addresses: 9000_0014 / 9000_8014 / 9001_0014 / 9001_8014 / 9002_0014 / 9002_8014 / 9003_0014 / 9003_8014

The Direct Mode RX Data FIFO register is used when in direct mode of operation, to read the RX Data FIFO.

Note: The Module Direct Mode RX FIFO Status register must be read before this register is read, to determine the valid number of bytes in the 32-bit access. The data is packed in little endian format.

Register



Register bit assignment

Bit(s)	Access	Mnemonic	Reset	Description
D31:00	R	RXD	N/A	RX Data FIFO Read register

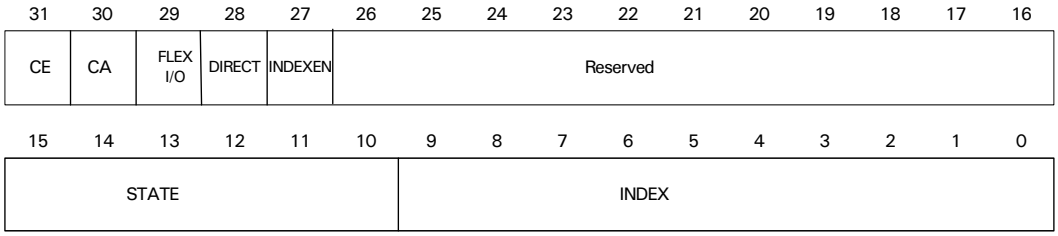
Downloaded from www.Manualslib.com manuals search engine

[*Module*] DMA TX Control

Addresses: 9000_0018 / 9000_8018 / 9001_0018 / 9001_8018 / 9002_0018 / 9002_8018 / 9003_0018

The DMA TX Control register contains control register settings for each transmit DMA channel.

Register



Register bit assignment

Bit(s)	Access	Mnemonic	Reset	Description
D31	R/W	CE	0x0	**Channel enable** 0 Disable DMA operation 1 Enable DMA operation
D30	R/W	CA	0x0	**Channel abort** When set, causes the current DMA operation to complete and closes the buffer. The DMA channel remains idle until this bit is cleared.
D29	R/W	FLEX I/O	0x0	0 DMA controlled by CPU 1 DMA controlled by flexible I/O module This bit is valid only for channels 0 and 1, which are assigned to flexible I/O module 0 and flexible I/O module 1.
D28	R/W	DIRECT	0x0	0 DMA mode 1 Direct access mode
D27	R/W	INDEXEN	0x0	0 Hardware will not use the INDEX field when in the idle state 1 Hardware will use the INDEX field when in the idle state
D26:16	N/A	Reserved	N/A	N/A
D15:10	R	STATE	0x0	DMA state machine status field
D09:00	R/W	INDEX	0x0	When the state machine is in the idle state, this register can be used to change the index. This field can be read at any time to determine the current index.

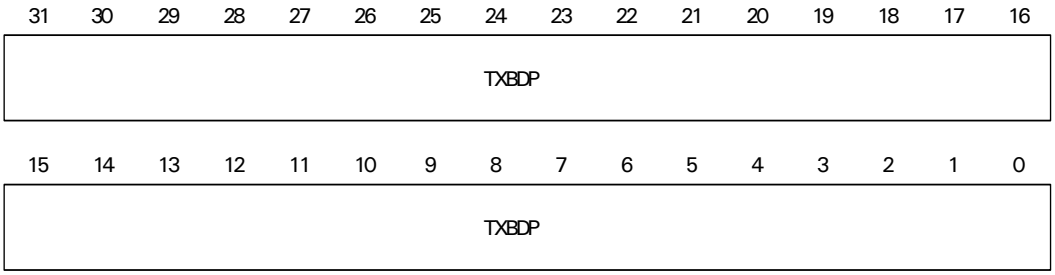
Downloaded from www.Manualslib.com manuals search engine

[*Module*] DMA TX Buffer Descriptor Pointer

Addresses: 9000_001C / 9000_801C / 9001_001C / 9001_801C / 9002_001C / 9002_801C / 9003_001C

The DMA TX Buffer Descriptor Pointer is the address of the first buffer descriptor for each DMA channel.

Register



Register bit assignment

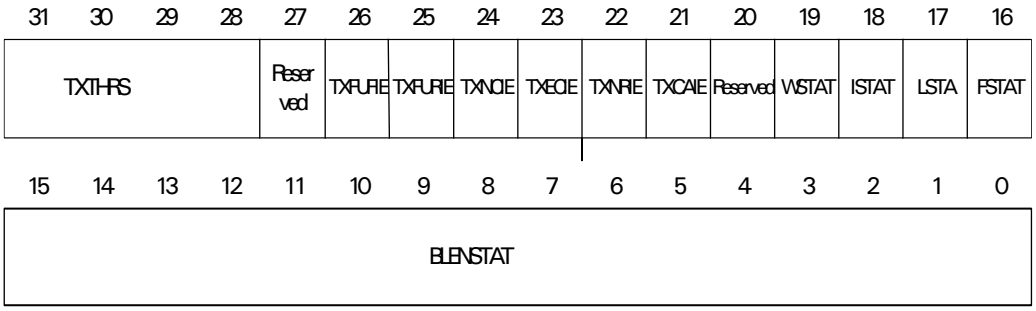
Bit(s)	Access	Mnemonic	Reset	Description
D31:00	R/W	TXBDP	0x0	The first buffer descriptor in the ring. Used when the W bit is found, which indicates the last buffer descriptor in the list.

[*Module*] TX Interrupt Configuration register

Addresses: 9000_0020 / 9000_8020 / 9001_0020 / 9001_8020 / 9002_0020 / 9002_8020 / 9003_0020

The TX Interrupt Configuration register allows system software to configure the interrupt from the I/O hub module transmit channel.

Register



Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

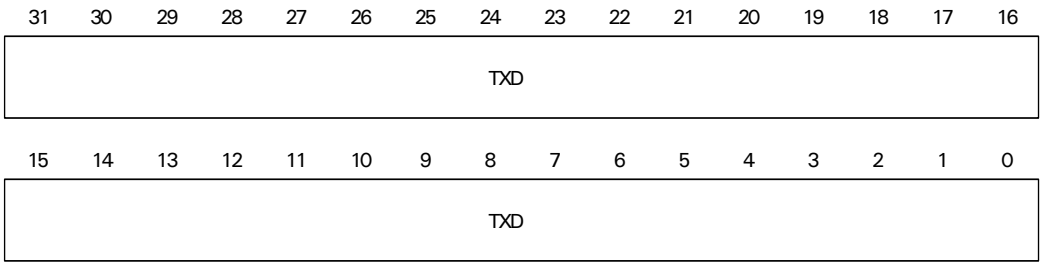
Bit(s)	Access	Mnemonic	Reset	Description
D31:28	R/W	TXTHRS	0xF	**TX FIFO threshold** An interrupt is generated when the FIFO level drops below this level.
D27	N/A	Reserved	N/A	N/A
D26	R/W	TXFUFIE	0x0	Enable the TXFUFIP interrupt.
D25	R/W	TXFSRIE	0x0	Enable the TXFSRIP interrupt.
D24	R/W	TXNCIE	0x0	Enable the NCIP interrupt.
D23	R/W	TXECIE	0x0	Enable the ECIP interrupt.
D22	R/W	TXNRIE	0x0	Enable the NRIP interrupt.
D21	R/W	TXCAIE	0x0	Enable the CAIP interrupt.
D20	N/A	Reserved	N/A	N/A
D19	R	WSTAT	0x0	Debug field, indicating the W bit is set in the current buffer descriptor.
D18	R	ISTAT	0x0	Debug field, indicating the I bit is set in the current buffer descriptor.
D17	R	LSTAT	0x0	Debug field, indicating the L bit is set in the current buffer descriptor.
D16	R	FSTAT	0x0	Debug field, indicating the F bit is set in the current buffer descriptor.
D15:00	R	BLENSTAT	0x0	Debug field, indicating the current byte count.

[*Module*] Direct Mode TX Data FIFO

Addresses: 9000_0028 / 9000_8028 / 9001_0028 / 9001_8028 / 9002_0028 / 9002_8028 / 9003_0028

The Direct Mode TX Data FIFO register is used when in direct mode of operation, to write the TX data FIFO. The write can be 8-, 16-, or 32-bit.

Register



Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

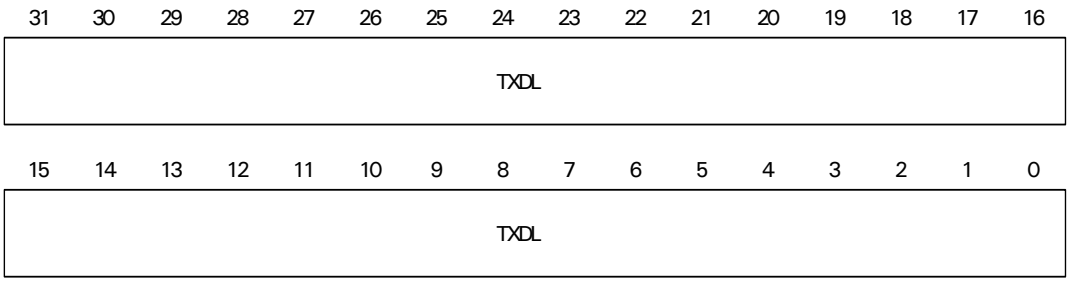
Bit(s)	Access	Mnemonic	Reset	Description
D31:00	W	TXD	0x0	TX Data FIFO Write register

[*Module*] Direct Mode TX Data Last FIFO

Addresses: 9000_002C / 9000_802C / 9001_002C /9001_802C / 9002_002C / 9000_802C / 9003_002C

The Direct Mode TX Data LAst FIFO register is used when in direct mode of operation, to write to the TX data FIFO and to cause a last status flag to be set for use by the peripheral. The write can be 8-, 16-, or 32-bit.

Register



Register bit assignment

Bit(s)	Access	Mnemonic	Reset	Description
D31:00	W	TXDL	0x0	TX Data with Last Status FIFO Write register.

Downloaded from www.Manualslib.com manuals search engine

Downloaded from www.Manualslib.com manuals search engine

Serial Control Module: UART

CHAPTER 10

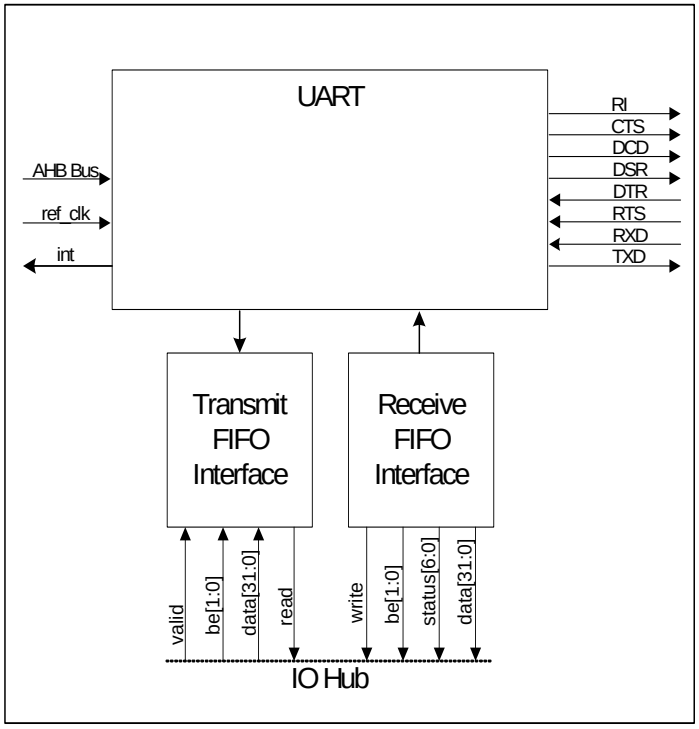
The processor ASIC supports four independent universal asynchronous receiver/transmitter (UART) channels (A through D). Each channel supports several modes, conditions, and formats.

Features

- DMA transfers to and from system memory
- Independent receive and transmit programmable bit-rate generators
- High speed data transfer up to 1.8432 Mbps
  - Programmable data format
    5 to 8 data bits
    Odd, even, or no parity
    1 or 2 stop bits
    MSB or LSB first
- Programmable channel modes
  - Normal
  - Local loopback
  - Remote loopback
- Modem control signal support
  - RTS, CTS, DTR, DSR, DCD, RI
- Maskable interrupt conditions
  - Receiver idle
  - Transmitter idle
  - Receive error conditions
  - Character gap timeout
  - Character match events
  - CTS, DSR, DCD, RI state change detection
- RS485 transceiver control signal
- Transmit FIFO bypass to force out a character

Downloaded from www.Manualslib.com manuals search engine

UART module structure



Normal mode operation

The UART achieves normal mode operation by programming the UART and Wrapper configuration registers.

Example configuration

This example shows a normal mode operation configuration for a hyperterminal application. Any field not specified in this table can be left at reset value.

Control register	Field	Value	Comment
UART Line Control register (0x10c)	DLAB	0x1	Enables access to baud rate registers
UART Baud Rate Divisor LSB (0x100)	DLR	0xC0	Set baud rate to 9600 bps MSB defaults to 0x0
UART Line Control register (0x10c)	DLAB	0x0	Disables access to baud rate registers
	WLS	0x3	8 bits per character
UART FIFO Control register (0x108)	FIFOEN	0x01	Enable RX and TX FIFOs

Downloaded from www.Manualslib.com manuals search engine

Control register	Field	Value	Comment
UART Interrupt Enable register (0x104)	ETBEI	0x1	Enable the Transmitter Holding Register Empty Interrupt. enables the Wrapper to write a transmit character to the UART.
Wrapper Configuration register	TX FLOW Software	1	TX Enabled
	RXEN	1	Enable Wrapper receive function
	TXEN	1	Enable Wrapper transmit function

Baud rate generator

The baud rate clock is generated by dividing the system reference clock by a programmable divisor; use this formula:

$$BR = CLK_{ref} / (BRD \times 16)$$

The default reference clock for the UARTs is the system reference clock input on x1_sys_osc. The UART reference clock optionally can be input on GPIO_A[3].

Baud rates

This table shows the baud rates achieved with CLK_{ref} set to 29.4912:

Divisor	Baud rate
1	1,843,299
2	921,600
4	460,800
8	230,400
16	115,200
32	57,600
48	38,400
64	28,800
96	19,200
128	14,400
192	9,600
384	4,800
768	2,400

Downloaded from www.Manualslib.com manuals search engine

Hardware-based flow control

The UART module provides expanded functionality for hardware-based flow control. The RTS signal normally indicates the state of the receive FIFO. The CTS signal normally halts the transmitter. With this UART module, the RI, CTS, DCD, or DSR signals can halt the transmitter. Program these features using the HWFLOW bits in the Wrapper Configuration register.

Character-based flow control (XON/XOFF)

Traditional character-based flow control requires the processor to match the flow control characters and control the transmitter accordingly. This UART module performs the character matching function in hardware and automatically updates the state of the transmitter, which allows character-based flow control to achieve the same response time as hardware-based flow control.

Example configuration

Configure the character-based flow control using at least two Receive Character Match registers and the Receive Character-Based Flow Control register. This table shows a sample configuration for a system transferring 8 data bits per character.

Control register	Field	Value	Comment
Receive Character Match Control Register 0	ENABLE	1	Enable character match
	MASK	0x00	Mask bits
	DATA	0x7e	Define character
Receive Character Match Control Register 1	ENABLE	1	Enable character match
	MASK	0x00	Mask bits
	DATA	0x81	Define character
Receive Character-Based Flow Control register	FLOW0	0x2	XON when matched
	FLOW1	0x3	XOFF when matched

Forced character transmission

UART provides a mechanism in which you can bypass data in the transmit FIFO with a specific character. The specified character is transmitted after the current

Downloaded from www.Manualslib.com manuals search engine

character completes, regardless of any flow control mechanism that might stall normal data transmission.

Use the Force Transmit Character Control register to program this operation.

Force character transmission procedure

These steps outline a single force character transmission operation:

1. Read the Force Transmit Character Control register and verify that the ENABLE field is 0. The Force Transmit Character Control register must not be written while the ENABLE field is 1.
2. Write a 1 to the ENABLE field and the required character to the CHAR field. This operation can be a single step.

Collecting feedback

Force character transmission completion status is available. It is up to you as to whether you want to collect feedback. If you do want to collect feedback, these are your options:

- Poll the ENABLE field in the Force Transmit Character Control register until it reads 0.
- Poll the FORCE field in the Interrupt Status register until it reads 1.
- Enable the FORCE interrupt by writing a 1 to the FORCE field in the Interrupt Enable register and servicing the interrupt when it occurs.

ARM wakeup on character recognition

The UART module provides a signal to the SCM module that can wake up the ARM processor. This signal is asserted when a specified character is received. Use the Receive Character Match Control registers and the ARM Wakeup Control register to implement the logic.

Example configuration

This table shows a sample configuration where the wakeup signal is asserted on reception of any character:

Control register	Field	Value	Comment
Receive Character Match Control Register 0	ENABLE	1	Enable character match
	MASK	0xff	Mask all bits
	DATA	0x00	Don’t care
ARM Wakeup Control register	ENABLE	1	Enable the function

Downloaded from www.Manualslib.com manuals search engine

Wrapper Control and Status registers

The configuration registers for UART module A start at 0x9001_1000, UART module B start at 0x9001_9000, UART module C start at 0x9002_1000, and UART module D start at 9002_9000.

Register address map

These are the configuration registers for UART module A. The configuration registers for other UART modules are the same, except they have different starting addresses.

Address	Register
9001_1000	Wrapper Configuration
9001_1004	Interrupt Enable
9001_1008	Interrupt Status
9001_100C	Receive Character GAP Control
9001_1010	Receive Buffer GAP Control
9001_1014	Receive Character Match Control 0
9001_1018	Receive Character Match Control 1
9001_101C	Receive Character Match Control 2
9001_1020	Receive Character Match Control 3
9001_1024	Receive Character Match Control 4
9001_1028	Receive Character-Based Flow Control
9001_102C	Force Transit Character Control
9001_1030	ARM Wakeup Control
9001_1034	Transmit Byte Count
9001_1038–9001_109C	
9001_1100 DLAB=0	UART Receive Buffer (read) UART Transmit Holding (write)
9001_1100 DLAB=1	UART Baud Rate Divisor LSB
9001_1104 DLAB=0	UART Baud Interrupt Enable
9001_1104 DLAB=1	UART Baud Rate Divisor MSB
9001_1108	UART Identification (read) UART FIFO Control (write)
9001_110C	UART Line Control
9001_1110	UART Modem Control

Downloaded from www.Manualslib.com manuals search engine

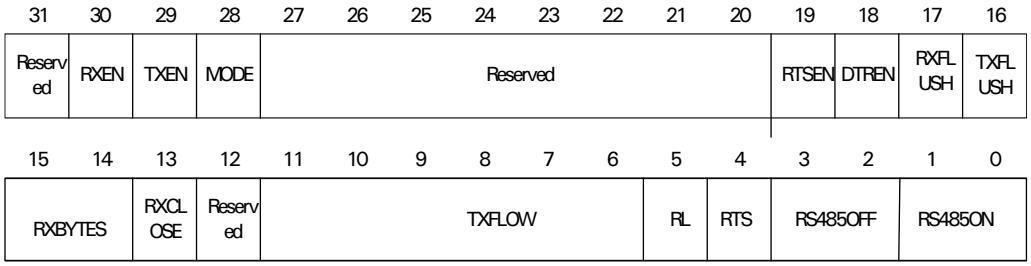
Address	Register
9001_1114	UART Line Status
9001_1118	UART Modem Status
9001_111C	UART Scratch

Wrapper Configuration register

Address: 9001_1000 / 9001_9000 / 9002_1000 / 9002_9000

This is the primary Wrapper Configuration register.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31	N/A	Reserved	N/A	N/A
D30	R/W	RXEN	0	0 Disable wrapper function 1 Enable wrapper to process receive characters
D29	R/W	TXEN	0	0 Disable transmitter function 1 Enable wrapper to process transmit characters
D28	R/W	MODE	0	Selects either UART or HDLC mode. This bit applies only to UART3. 0 UART mode 1 HDLC mode
D27:20	N/A	Reserved	N/A	N/A
D19	R/W	RTSEN	0	Indicates which signal is output: RTS or RS485 transceiver control. 0 RTS 1 RS485 transceiver control
D18	R/W	DTREN	0	Indicates which signal is output: DTR or TX baud clock. 0 DTR 1 TX baud clock

Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Reset	Description
D17	R/W	RXFLUSH	0	Resets the contents of the 64-byte RXFIFO. Write a 1, then a 0 to reset the FIFO.
D16	R/W	TXFLUSH	N/A	Resets the contents of the 64-byte TX FIFO. Write a 1, then a 0 to reset the FIFO.
D15:14	R	RXBYTES	00	Indicates how many bytes are pending in the wrapper. The wrapper writes to the RX FIFO only when 4 bytes are received or a buffer close event occurs, such as a character gap timeout, character match, or error.
D13	R/W	RXCLOSE	0	Allows software to close a receive buffer. Hardware clears this bit when the buffer has been closed. 0 Idle or buffer already closed 1 Software initiated buffer close
D12	N/A	Reserved	N/A	N/A
D11:06	R/W	TXFLOW	010000	Selects which signals are routed to the UART for hardware flow control. Transmit data is halted when the selected signal is deasserted. **[0] CTS** 0 CTS disabled 1 CTS enabled **[1] DCD** 0 DCD disabled 1 DCD enabled **[2] DSR** 0 DSR disabled 1 DSR enabled **[3] RI** 0 RI disabled 1 RI enabled **[4] Software** 0 TX disabled 1 TX enabled **[5] Receive character-based flow control** 0 Disabled 1 Enabled
D05	R/W	RL	0	**Remote loopback** Provides an internal remote loopback feature. When the RL field is set to 1, the receive serial data signal is connected to the transmit serial data signal. A local loopback is provided in the UART.
D04	R/W	RTS	0	**RTS control** 0 Controlled directly by UART 1 Deasserted when RX FIFO is half full

Downloaded from www.Manualslib.com manuals search engine

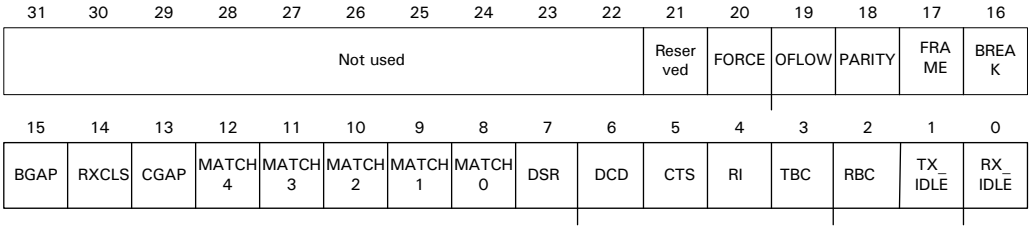
Bits	Access	Mnemonic	Reset	Description
D03:02	R/W	RS485OFF	00	**RS485 transceiver deassertion control** In bit times after the stop bit period 00 0 01 1 10 1.5 11 2
D01:00	R/W	RS485ON	00	**RS485 transceiver assertion control** In bit times before the falling edge of the start bit 00 0 01 1 10 1.5 11 2

Interrupt Enable register

Address: 9001_1004 / 9001_9004 / 9002_1004 / 9002_9004

Use the Interrupt Enable register to enable interrupt generation on specific events. Enable the interrupt by writing a 1 to the appropriate bit field(s).

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:22	R/W	Not used	0	Write this field to 0.
D21	R/W	Reserved	0	Always write to 0.
D20	R/W	FORCE	0	**Enable force complete** Enables interrupt generation when a force character transmission operation has completed.

Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Reset	Description
D19	R/W	OFLOW	0	**Enable overflow error** Enables interrupt generation if the 4-character FIFO in the UART overflows. Note: This should not happen in a properly configured system.
D18	R/W	PARITY	0	**Enable parity error** Enables interrupt generation when a character is received with a parity error.
D17	R/W	FRAME	0	**Enable frame error** Enables interrupt generation when a character is received with a framing error.
D16	R/W	BREAK	0	**Enable line break** Enables interrupt generation when a line break condition occurs.
D15	R/W	BGAP	0	**Enable buffer gap** Enables interrupt generation when a buffer gap timeout event occurs.
D14	R/W	RXCLS	0	**Software receive close** Enables interrupt generation when software forces a buffer close.
D13	R/W	CGAP	0	**Enable character gap** Enables interrupt generation when a character gap timeout event occurs.
D12	R/W	MATCH4	0	**Enable character match4** Enables interrupt generation when a receive character match occurs against the Receive Match Register 4.
D11	R/W	MATCH3	0	**Enable character match3** Enables interrupt generation when a receive character match occurs against the Receive Match Register 3.
D10	R/W	MATCH2	0	**Enable character match2** Enables interrupt generation when a receive character match occurs against the Receive Match Register 2.
D09	R/W	MATCH1	0	**Enable character match1** Enables interrupt generation when a receive character match occurs against the Receive Match Register 1.
D08	R/W	MATCH0	0	**Enable character match0** Enables interrupt generation when a receive character match occurs against the Receive Match Register 0.
D07	R/W	DSR	0	**Enable data set ready** Enables interrupt generation whenever a state change occurs on input signal DSR.

Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Reset	Description
D06	R/W	DCD	0	**Enable data carrier** Enables interrupt generation whenever a stat change occurs on input signal DCD.
D05	R/W	CTS	0	**Enable clear to send** Enables interrupt generation whenever a state change occurs on input signal CTS.
D04	R/W	RI	0	**Enable ring indicator** Enables interrupt generation whenever a state change occurs on input signal RI.
D03	R/W	TBC	0	**Enable transmit buffer close** Enables interrupt generation when the UART transmit FIFO indicates to the UART transmitter that a byte corresponds to a buffer close event.
D02	R/W	RBC	0	**Enable receive buffer close** Enables interrupt generation whenever a buffer close event is passed from the UART receiver to the receive FIFO. These are the UART receive buffer close events: 1 Receive character match 2 Receive character gap timeout 3 Receive line break 4 Receive framing error 5 Receive parity error
D01	R/W	TX_IDLE	0	**Enable transmit idle** Enables interrupt generation whenever the transmitter moves from the active state to the idle state. This indicates that the transmit FIFO is empty and the transmitter is not actively shifting out data.
D00	R/W	RX_IDLE	0	**Enable receive idle** Enables interrupt generation whenever the receiver moves from the active state to the idle state. If a start bit is not received after a stop bit, the receiver enters the idle state.

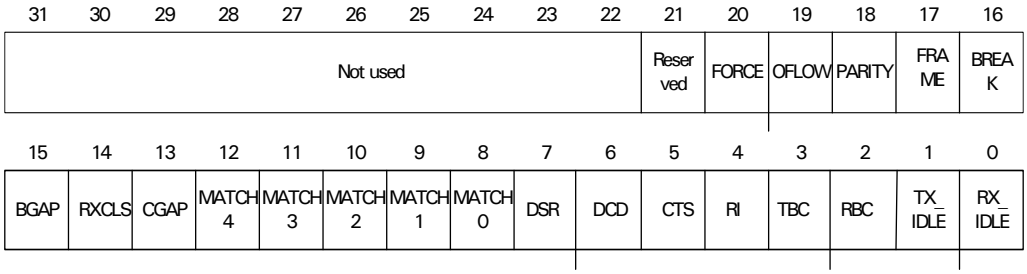
Interrupt Status register

Address: 9001_1008 / 9001_9008 / 9002_1008 / 9002_9008

The Interrupt Status register provides status about UART events. All events are indicated by reading a 1 and are cleared by writing a 1.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:22	R/W	Not used	0	Write this field to 0.
D21	R/W1TC	Reserved	0	**UART interrupt** Indicates that the UART has generated an interrupt.
D20	R/W1TC	FORCE	0	**Force complete** Indicates that a force character transmission operation has completed.
D19	R/W1TC	OFLOW	0	**Enable overflow error** Indicates that an overflow occurred in the UART's 4-character FIFO. Note: This should not happen in a properly configured system.
D18	R/W1TC	PARITY	0	**Parity error** Indicates that at least one character has been received with a parity error.
D17	R/W1TC	FRAME	0	**Frame error** Indicates that at least one character has been received with a framing error.
D16	R/W1TC	BREAK	0	**Line break** Indicates that a line break condition has occurred.
D15	R/W1TC	BGAP	0	**Buffer gap** Indicates that a buffer gap timeout event has occurred.
D14	R/W1TC	RXCLS	0	**Software receive close** Indicates a software-initiated buffer close has completed.
D13	R/W1TC	CGAP	0	**Character gap** Indicates that a character gap timeout event has occurred.
D12	R/W1TC	MATCH4	0	**Character match4** Indicates that a receive character match has occurred against the Receive Match Register 4.

Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Reset	Description
D11	R/W1TC	MATCH3	0	**Character match3** Indicates that a receive character match has occurred against the Receive Match Register 3.
D10	R/W1TC	MATCH2	0	**Character match2** Indicates that a receive character match has occurred against the Receive Match Register 2.
D09	R/W1TC	MATCH1	0	**Character match1** Indicates that a receive character match has occurred against the Receive Match Register 1.
D08	R/W1TC	MATCH0	0	**Character match0** Indicates that a receive character match has occurred against the Receive Match register 0.
D07	R/W1TC	DSR	0	**Data set ready** Indicates that a state change has occurred on input signal DSR.
D06	R/W1TC	DCD	0	**Data carrier detect** Indicates that a state change has occurred in input signal DCD.
D05	R/W1TC	CTS	0	**Clear to send** Indicates that a state change has occurred on input signal CTS.
D04	R/W1TC	RI	0	**Ring indicator** Indicates that a state change has occurred on input signal RI.
D03	R/W1TC	TBC	0	**Transmit buffer close** Indicates that transmission of the last byte in a transmit buffer has completed.
D02	R/W1TC	RBC	0	**Receive buffer close** Indicates that a UART receive buffer close condition has occurred. These are UART receive buffer close events: 1 Receive character match 2 Receive character gap timeout 3 Receive line break 4 Receive framing error 5 Receive parity error

Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Reset	Description
D01	R/W1TC	TX_IDLE	0	**Transmit idle** Indicates that the transmitter has moved from the active state to the idle state. The transmitter moves from the active state to the idle state when the transmit FIFO is empty and the transmitter is not actively shifting out data.
D00	R/W1TC	RX_IDLE	0	**Receive idle** Indicates that the receiver has moved from the active state to the idle state. The receiver moves from the active state to the idle state when a start bit has not been received after the previous stop bit.

Receive Character GAP Control register

Address: 9001_100C / 9001_900C / 9002_100C / 9002_900C

The Receive Character GAP Control register configures the receive character gap control logic.

Register

REGISTER

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31	R/W	ENABLE	0	**Enable receive character gap timer** Write a 1 to this field to enable the receive character gap timer.
D30:25	R/W	Not used	0x0	**Write this field to 0.**
D24:00	R/W	VALUE	0	**Value** Defines the period between receiving the stop bit and asserting the character gap timeout event. Use this equation to compute the required divisor value: N = ((FCLK * gap+period) - 1) F_{CLK} = Nominal 29.4912 MHz gap_period = Desired character gap period A reasonable setting is 10 bit periods one character plus the start and stop bits. Given a data rate of 115,200bps, the desired period is 86.8us and the timeout value is 2559_d.

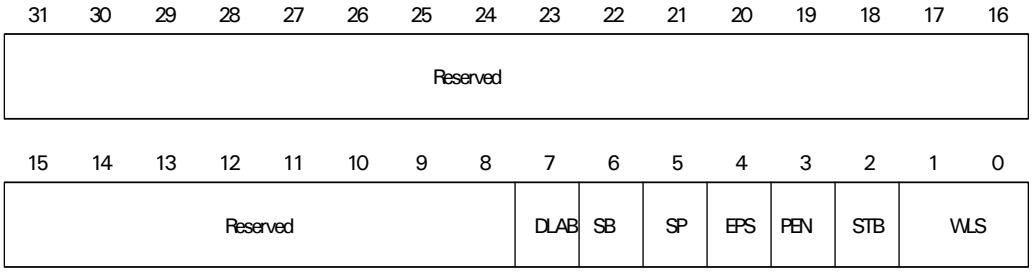
Downloaded from www.Manualslib.com manuals search engine

Receive Buffer GAP Control register

Address: 9001_1010 / 9001_9010 / 9002_1010 / 9002_9010

The Receive Buffer GAP Control register configures the receive buffer gap control logic. The buffer gap timer starts when the first character in a new buffer is received.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31	R/W	ENABLE	0	**Enable transmit bit rate generation** Write a 1 to enable the transmit bit rate generator.
D30:25	R/W	Not used	0x0	**Write this field to 0.**
D24:00	R/W	VALUE	0	**Value** Defines the period between receiving the stop bit and asserting the buffer gap timeout event. Use this equation to compute the required divisor value: N = ((FCLK * gap_period) - 1) F_{CLK} = Nominal 29.4912 Mhz gap_period = Desired buffer gap period A reasonable setting is 64 character or 640 bit periods. GIven a data rate of 115,200 bps, the desired period is 5.55ms and the timeout value is $163,839_d$.

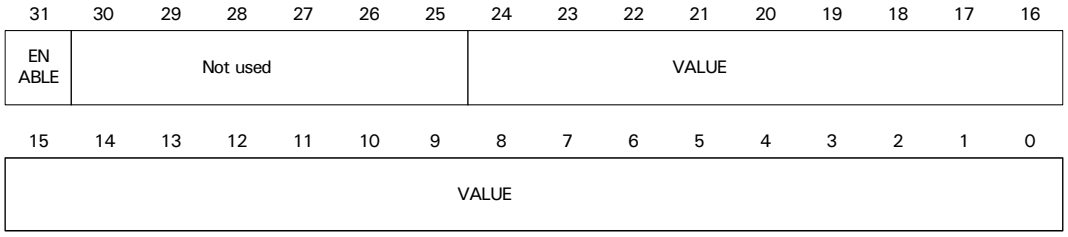
Receive Character Match Control register

Addresses: 9001_1014 / 9001_1018 / 9001_901C / 9001_9020 / 9001_1024 / 9001_9014 / 9001_9018 / 9001_901C / 9001_9020 / 9001_9024 / 9002_1014 / 9002_1018 / 9002_101C / 9002_1020 / 9002_1024 / 9002_9014 / 9002_9018 / 9002_901C / 9002_9020 / 9002_9024

Downloaded from www.Manualslib.com manuals search engine

The Receive Character Match Control registers configure the receive character match control logic. Each UART module has five Receive Character Match Control registers.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31	R/W	ENABLE	0	**Enable character match** Write a 1 to enable the receive character match control logic.
D30:24	R	Not used	0x0	Write this field to 0.
D23:16	R/W	MASK	0x0	**Mask** Allows you to not include specific bits in the receive character match operation. Writing 1 masks off the bit in the specified position. Bit positions that are not used should always be masked. For example, bit positions 9 through 12 should always be masked for 8-bit characters.
D15:08	R	Not used	0x0	Write this field to 0.
D07:00	R/W	DATA	0x0	**Data** Allows you to specify the receive characters to match against.

Receive Character-Based Flow Control register

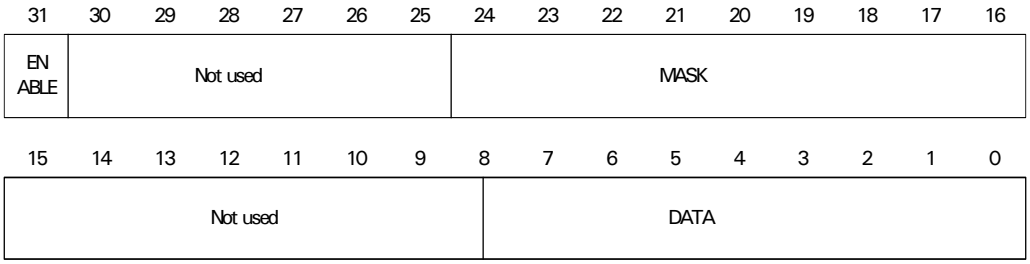
Address: 9001_1028 / 9001_9028 / 9002_1028 / 9002_9028

The Receive Character-Based Flow Control register lets you define the UART module's receive character-based flow control operation. Use this register in conjunction with the Receive Character Match Control registers to define the flow control characters. If enabled, this function's output is wired to the UART module instead of the CTS signal.

Downloaded from www.Manualslib.com manuals search engine

Caution: Be aware that if multiple matches occur, an XOFF assertion will supersede an XON assertion.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:11	R	Not used	0	Write this field to 0.
D10	R	FLOW_STATE	0x0	**Flow control state** 0 Hardware initiated XON 1 Hardware initiated XOFF
D09:08	R/W	FLOW4	0	**Flow control enable** Allows you to define flow characteristics using the DATA and MASK fields on the Receive Character Match Control Register 4. Note: The ENABLE field has no effect on the flow control logic. The flow control is defined as shown: 0x Disabled 10 Change the FLOW_STATE field to XON upon match 11 Change the FLOW_STATE field to XOFF upon match
D07:06	R/W	FLOW3	0	**Flow control enable** Allows you to define flow characteristics using the DATA and MASK fields on the Receive Character Match Control Register 3. Note: The ENABLE field has no effect on the flow control logic. The flow control is defined as shown: 0x Disabled 10 Change the FLOW_STATE field to XON upon match 11 Change the FLOW_STATE field to XOFF upon match

Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Reset	Description
D05:04	R/W	FLOW2	0	**Flow control enable** Allows you to define flow characteristics using the DATA and MASK fields on the Receive Character Match Control Register 2. Note: The ENABLE field has no effect on the flow control logic. The flow control is defined as shown: 0x Disabled 10 Change the FLOW_STATE field to XON upon match 11 Change the FLOW_STATE field to XOFF upon match
D03:02	R/W	FLOW1	0	**Flow control enable** Allows you to define flow characteristics using the DATA and MASK fields on the Receive Character Match Control Register 1. Note: The ENABLE field has no effect on the flow control logic. The flow control is defined as shown: 0x Disabled 10 Change the FLOW_STATE field to XON upon match 11 Change the FLOW_STATE field to XOFF upon match
D01:00	R/W	FLOW0	0	**Flow control enable** Allows you to define flow characteristics using the DATA and MASK fields on the Receive Character Match Control Register 0. Note: The ENABLE field has no effect on the flow control logic. The flow control is defined as shown: 0x Disabled 10 Change the FLOW_STATE field to XON upon match 11 Change the FLOW_STATE field to XOFF upon match

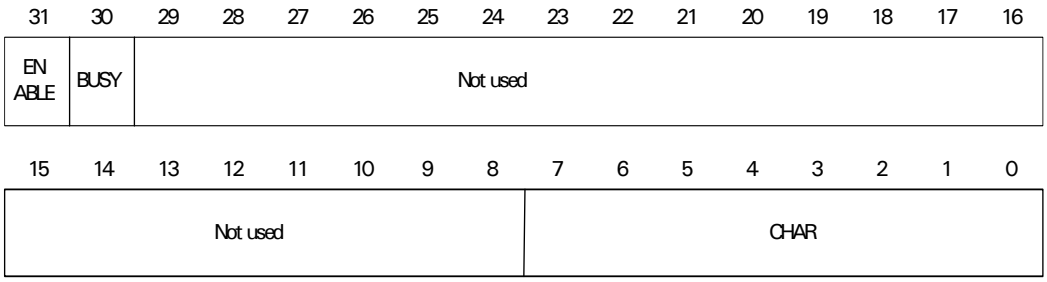
Force Transmit Character Control register

Address: 9001_102C / 9001_902C / 9002_102C / 9002_902C

Use the Force Transmit Character Control register to override the normal flow of transmit data.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31	R/W	ENABLE	0	**Force transmit enable** Use this field to force the transmitter to send the character specified in the CHAR field (D07:00). All user-specified rules, such as bit order, parity, or number of stop bits, are enforced. Write a 1 to enable this field. Hardware clears the field once the character has been transmitted. Writing a 1 to this field when it is already a 1 has unpredictable results. Note: Writing a 1 to this field also clears the FORCE field in the Interrupt Status register.
D30	R	BUSY	0	**Read-only busy** Reading a 1 indicates that the force operation you initiated is in progress.
D29:08	R	Not used	0	Write this field to 0.
D07:00	R/W	CHAR	0	**Force character** Defines the character that is forced out of the transmitter.

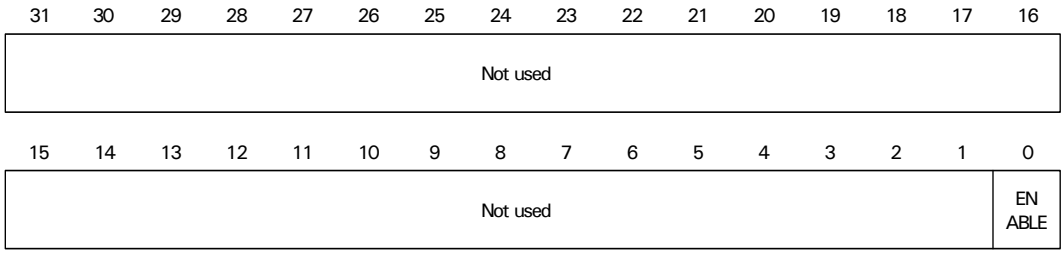
ARM Wakeup Control register

Address: 9001_1030 / 9001_9030 / 9002_1030 / 9002_9030

Use the ARM Wakeup Control register to enable the ARM wakeup control logic.

Downloaded from www.Manualslib.com manuals search engine

Register



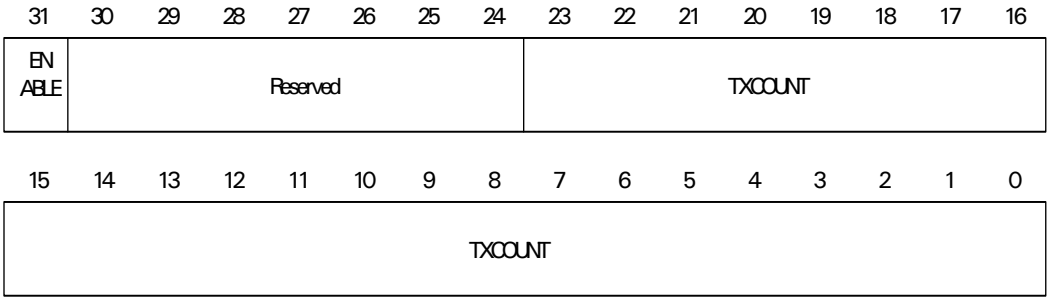
Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:01	R	Not used	0	Write this field to 0.
D00	R/W	ENABLE	0	**Enable** Write a 1 to this field to enable ARM wakeup control logic.

Transmit Byte Count

Address: 9001_1034 / 9001_9034 / 9002_1034 / 9002_9034

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31	R/W	ENABLE	0	Enables and resets the transmit byte counter: 0 Transmit byte count disabled and reset 1 Transmit byte enabled
D30:24	N/A	Reserved	N/A	N/A
D23:00	R	TXCOUNT	0	This counter is incremented after bytes are transmitted.

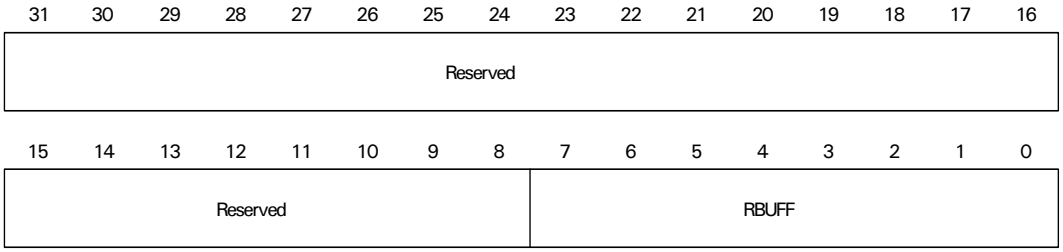
Downloaded from www.Manualslib.com manuals search engine

UART Receive Buffer

Address: 9001_1100 / 9001_9100 / 9002_1100 / 9002_9100, DLAB = 0, Read

UART Receive Buffer is used for diagnostic purposes only.

Register



Register bit assignment

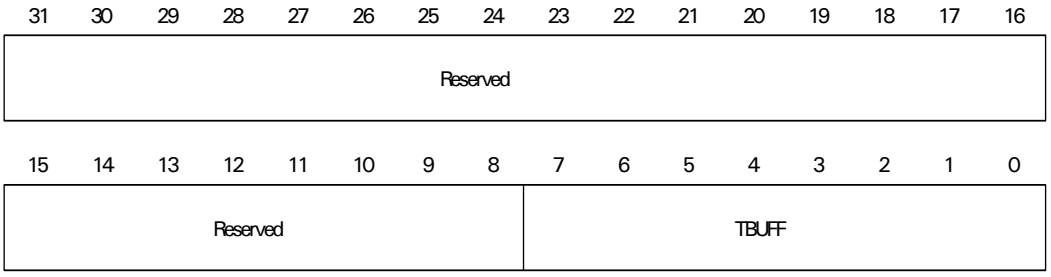
Bits	Access	Mnemonic	Reset	Description
D31:08	N/A	Reserved	N/A	N/A
D07:00	R	RBUFF	0	Receiver data bits

UART Transmit Buffer

Address: 9001_1100 / 9001_9100 / 9002_1100 / 9002_9100, DLAB = 0, Write

UART Transmit Buffer is used for diagnostic purposes only.

Register



Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

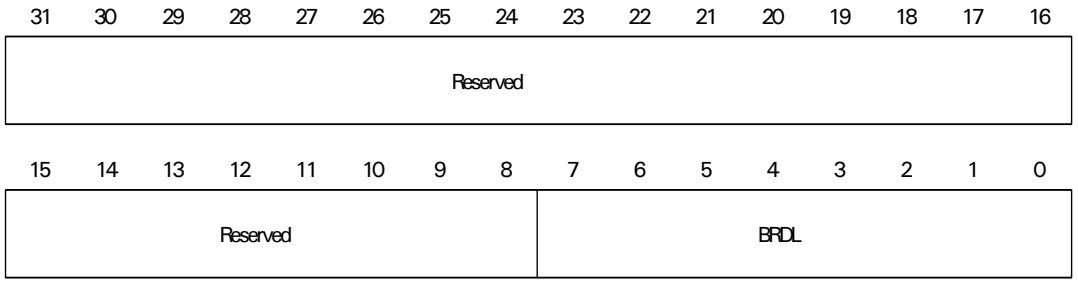
Bits	Access	Mnemonic	Reset	Description
D31:08	N/A	Reserved	N/A	N/A
D07:00	W	TBUFF	0	Transmitter data bits

UART Baud Rate Divisor LSB

Address: 9001_1100 / 9001_9100 / 9002_1100 / 9002_9100, DLAB = 1

UART Baud Rate Divisor sets bits 07:00 of the baud rate generator divisor.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:08	N/A	Reserved	N/A	N/A
D07:00	R/W	BRDL	0x1	Bits 07:00 of the baud rate generator divisor

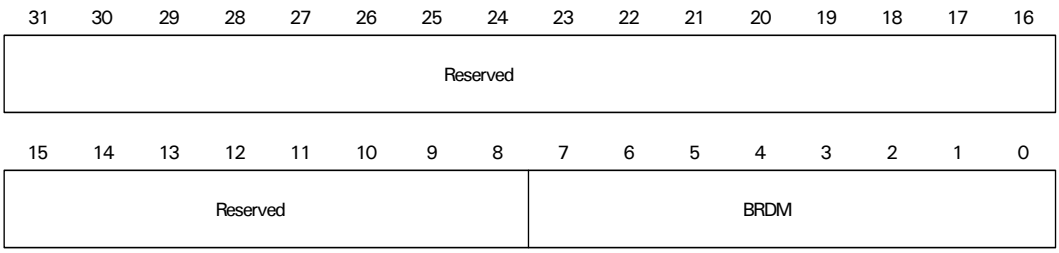
UART Baud Rate Divisor MSB

Address: 9001_1104 / 9001_9104 / 9002_1104 / 9002_9104, DLAB = 1

UART Baud Rate Divisor sets bits 15:08 of the baud rate generator divisor.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

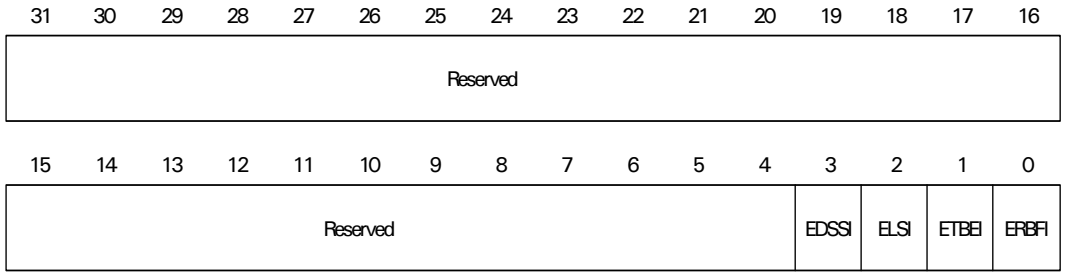
Bits	Access	Mnemonic	Reset	Description
D31:08	N/A	Reserved	N/A	N/A
D07:00	R/W	BRDM	0	Bits 15:08 of the baud rate generator divisor

UART Interrupt Enable register

Address: 9001_1104 / 9001_9104 / 9002_1104 / 9002_9104, DLAB = 0

The UART Interrupt Enable register selects the source of the interrupt from the UART. Note that only bit ETBEI (bit 01) must be set for normal operation. All other bits are for diagnostic purposes only.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:04	N/A	Reserved	N/A	N/A
D03	R/W	EDSSI	N/A	**Enables modem status interrupt** 0 Disabled 1 Enabled

Downloaded from www.Manualslib.com manuals search engine

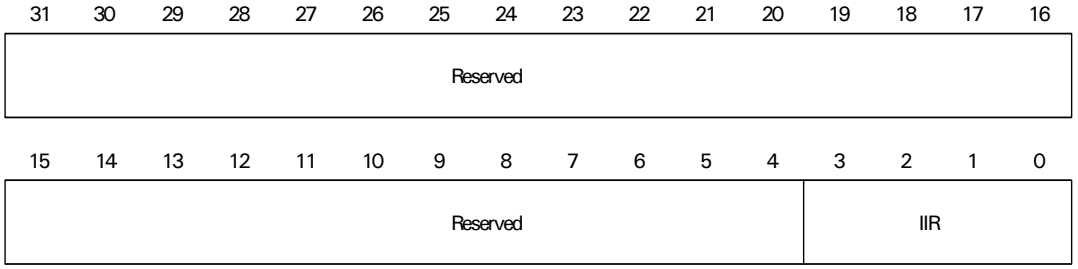
Bits	Access	Mnemonic	Reset	Description
D02	R/W	ELSI	0	**Enables receive line status interrupt** 0 Disabled 1 Enabled
D01	R/W	ETBEI	0	**Enables transmit holding register empty interrupt** 0 Disabled 1 Enabled
D00	R/W	ERBFI	0	**Enables receive data available interrupt** 0 Disabled 1 Enabled

UART Interrupt Identification register

Address: 9001_1108 / 9001_9108 / 9002_1108 / 9002_9108, Read

The UART Interrupt Identification register reads the source of the interrupt from the UART. This register is for diagnostic purposes only.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:04	N/A	Reserved	N/A	N/A
D03:00	R	IIR	N/A	**Interrupt identification** 0110 Receiver line status error 0100 Receive data available 0010 Transmit holding register empty 0000 Modem status

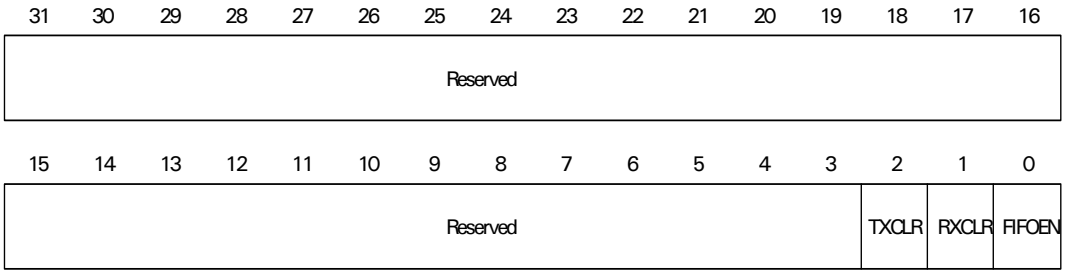
Downloaded from www.Manualslib.com manuals search engine

UART FIFO Control register

Address: 9001_1108 / 9001_9108 / 9002_1108 / 9002_9108, Write

The UART FIFO Control register controls the RX and TX 4-byte FIFOs. Note that only the FIFOEN bit (bit 01) should be set; all other bits are for diagnostic purposes only.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:03	N/A	Reserved	N/A	N/A
D02	W	TXCLR	0	**Clear all bytes in the TX FIFO** 0 Normal operation 1 TX FIFO cleared
D01	W	RXCLR	0	**Clear all bytes in the RX FIFO** 0 Normal operation 1 RX FIFO cleared
D00	W	FIFOEN	0	**Enable the TX and RX FIFO** 0 RX and TX FIFO disabled 1 RX and TX FIFO enabled

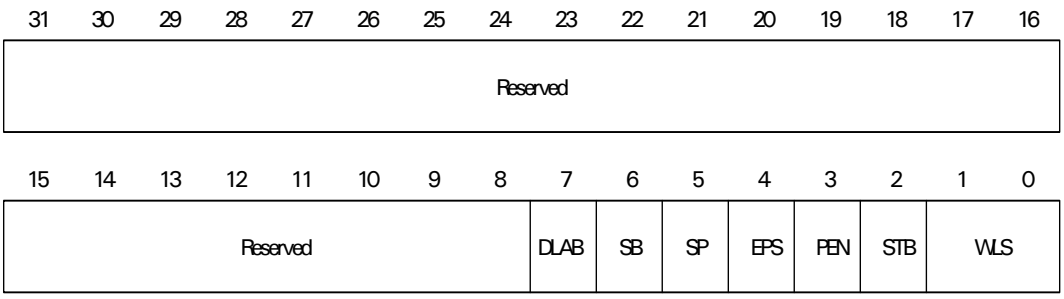
UART Line Control register

Address: 9001_110C / 9001_910C / 9002_110C / 9002_910C

The UART Line Control register controls the UART settings.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:08	N/A	Reserved	N/A	N/A
D07	R/W	DLAB	0	**Divisor latch access bit** 0 Disabled 1 Enabled. Enables the Baud Rate Divisor MSB and LSB registers to be configured.
D06	R/W	SB	0	**Set break, if set TX data is set to 0** 0 Disabled 1 Enabled
D05	R/W	SP	0	**Stick parity, operates as follows** ■ When set bits 04:03 = 11, parity bit always set to 0 ■ When set bits 04:03 = 00, parity bit always set to 1 0 Disabled 1 Enabled
D04	R/W	EPS	0	**Parity select** 0 Odd parity 1 Even parity
D03	R/W	PEN	0	**Parity enable** 0 Parity disabled 1 Parity enabled
D02	R/W	STB	0	**Number of stop bits** 0 1 stop bit 1 1.5 stop bits (WLS = 00) 2 stop bits (all other WLS settings)
D01:00	R/W	WLS	0	**Word length select** 00 5 bits 01 6 bits 10 7 bits 11 8 bits

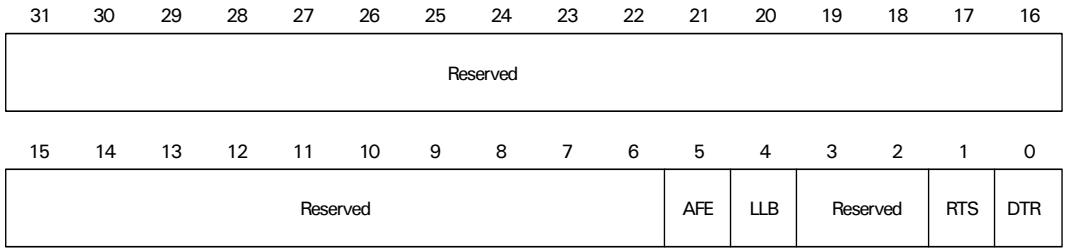
Downloaded from www.Manualslib.com manuals search engine

UART Modem Control register

Address: 9001_1110 / 9001_9110 / 9002_1110 / 9002_9110

The UART Modem Control register controls the modem signals.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:06	N/A	Reserved	N/A	N/A
D05	R/W	AFE	0	**Automatic flow control** 0 RTS controlled by bit 1 (RTS) 1 RTS controlled by 4-byte RX FIFO status
D04	R/W	LLB	0	**Local loopback enable bit** TX data looped back to RX data 0 Disabled 1 Enabled
D03:02	N/A	Reserved	N/A	N/A
D01	R/W	RTS	0	Controls the Request to Send (RTS) output 0 RTS = 1 1 RTS = 0
D00	R/W	DTR	0	Controls the Data Terminal Ready (DTR) output 0 DTR = 1 1 DTR = 0

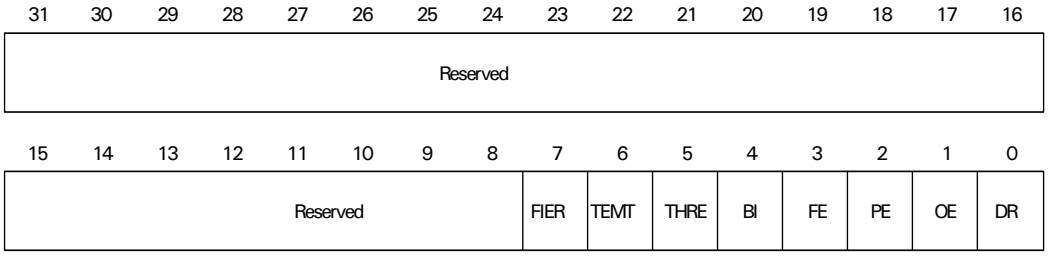
UART Line Status register

Address: 9001_1114 / 9001_9114 / 9002_1114 / 9002_9114

The UART Line Status register reads the line status register. This register is used for diagnostic purposes only.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:08	N/A	Reserved	N/A	N/A
D07	R	FIER	N/A	**RX FIFO error** Indicates at least one parity, framing, or break error in the RX FIFO.
D06	R	TEMT	N/A	Transmit holding and shift registers empty
D05	R	THRE	N/A	Transmit holding register empty
D04	R	BI	N/A	**Break indicator** The receiver found a line break.
D03	R	FE	N/A	**Framing error** The receiver found a framing error.
D02	R	PE	N/A	**Parity error** The receiver found a parity error.
D01	R	OE	N/A	**Overrun error** The RX FIFO experienced an overrun.
D00	R	DR	N/A	**Data ready** Indicates a data byte is ready in the FIFO.

UART Modem Status register

Address: 9001_1118 / 9001_9118 / 9002_1118 / 9002_9118

The UART Modem Status register reads the modem status register. This register is used for diagnostic purposes only.

Downloaded from www.Manualslib.com manuals search engine

Register

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:08	N/A	Reserved	N/A	N/A
D07	R	DCD	N/A	Reflects the status of the data carrier detect input.
D06	R	RI	N/A	Reflects the status of the ring indicator.
D05	R	DSR	N/A	Reflects the status of the data set ready input.
D04	R	CTS	N/A	Reflects the status of the clear to send input.
D03	R	DDCD	N/A	**Delta DCD indicator** Indicates that an edge was found on DCD since the last time the register was read.
D02	R	TERI	N/A	**Trailing edge of RI indicator** Indicates that RI has changed from a 0 to a 1.
D01	R	DDSR	N/A	**Delta DSR indicator** Indicates that an edge was found on DSR since the last time the register was read.
D00	R	DCTS	N/A	**Delta CTS indicator** Indicates that an edge was found on CTS since the last time the register was read.

Downloaded from www.Manualslib.com manuals search engine

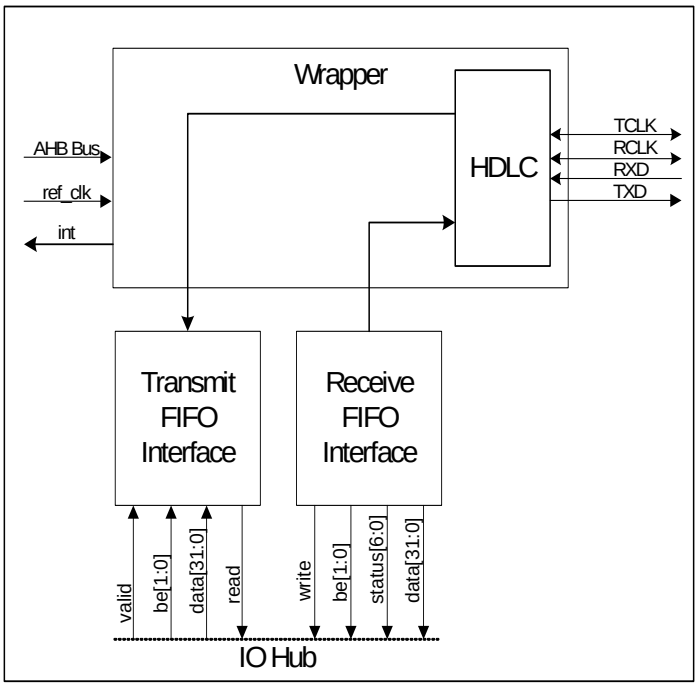
Downloaded from www.Manualslib.com manuals search engine

Serial Control Module: HDLC

CHAPTER 11

The HDLC module allows full-duplex synchronous communication. Both the receiver and transmitter can select either an internal or external clock. The HDLC module encapsulates data within opening and closing flags, and sixteen bits of CRC precedes the closing flag. All information between the opening and closing flag is *zero-stuffed;* that is, if five consecutive ones occur, independent of byte boundaries, a zero is automatically inserted by the transmitter and automatically deleted by the receiver. This allows a flag byte (07Eh) to be unique within a serial stream. The standard CRC-CCITT polynomial $(x16 + x12 + x5 + 1)$ is implemented, with the generator and checker preset to all ones.

HDLC module structure



Receive and transmit operations

Both receive and transmit operations are essentially automatic.

Downloaded from www.Manualslib.com manuals search engine

Receive operation

In the receiver, each byte is marked with status to indicate end-of-frame, short frame, and CRC error. The receiver automatically synchronizes on flag bytes, and presets the CRC checker accordingly. If the current receive frame is not needed (for example, because it is addressed to a different station), a flag search command is available. The flag search command forces the receiver to ignore the incoming data stream until another flag is received.

Transmit operation

In the transmitter, the CRC generator is preset and the opening flag transmitted automatically after the first byte is written to the transmitter buffer. The CRC an the closing flag are transmitted after the byte that is written to the buffer through the Address register. If no CRC is required, writing the last byte of the frame to the Long Stop register automatically appends a closing flag after the last byte.

Transmitter underflow

If the transmitter underflows, either an abort or a flag is transmitted, under software control. There is a command available to send the abort pattern (seven consecutive ones) if a transmit frame needs to be aborted prematurely. The abort command takes effect on the next byte boundary and causes an FEh (a zero followed by seven ones) transmission, after which the transmitter sends the idle line condition. The abort command also purges the transmit FIFO The idle line condition can be either flags or all ones.

Clocking

A 15-bit divider circuit provides the clocking for the HDLC module. This clock is sixteen times the data rate. The receiver uses a digital phase locked loop (DPLL) to generate a synchronized receive clock for the incoming data stream. The HDLC module also allows for an external 1x (same speed as the data rate) clock for both the receiver and the transmitter.

HDLC receive and transmit clocks can be input or output. When using an external clock, the maximum data rate is one-sixth of the 29.4912 MHz reference clock rate, or 4.9152 Mbps.

Bits

The transmitter cannot send an arbitrary number of bits, but only a multiple of bytes. The receiver, however, can receive frames of any bit length. If the last “byte” in the frame is not eight bits, the receiver sets a status flag that is buffered along with this last byte. Software then uses the table shown next to determine the number of valid data bits in this last “byte.” Note that the receiver transfers all bits

Downloaded from www.Manualslib.com manuals search engine

between the opening and closing flags, except for the inserted zeroes, to the receiver data buffer.

Last byte bit pattern table

Last byte bit pattern	Valid data
bbbbbbb0	7
bbbbbb01	6
bbbbb011	5
bbbb0111	4
bbb01111	3
bb011111	2
b0111111	1

Data encoding

The HDLC module provides several types of data encoding:

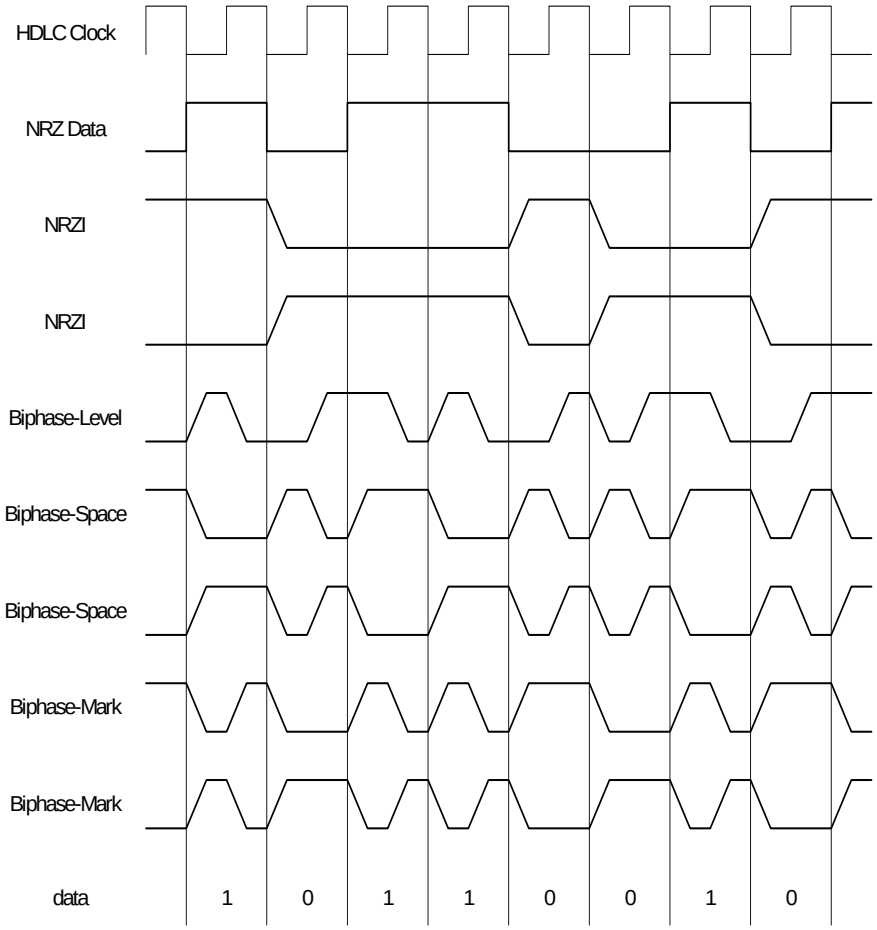
- Normal NRZ
- NRZI
- Biphase-Level (Manchester)
- Biphase-Space (FM0)
- Biphase-Mark (FM1)

Encoding examples

This figure shows examples of the data encoding types.

- In NRZI, Biphase-Space and Biphase-Mark, the signal level does not convey information. The placement of the transitions determine the data.
- In Biphase-Level, the polarity of the transmission determines the data.

Downloaded from www.Manualslib.com manuals search engine



Digital phase-locked-loop (DPLL) operation: Encoding

In the HDLC module, the internal clock comes from the output of the dedicated divider. The divider output is divided by 16 to form the transmit clock and is fed to the DPLL to form the receive clock. The DPLL basically is a divide-by-16 counter that uses the transition timings on the receive data stream to adjust its count. The DPLL adjusts the count so the DPLL output is placed properly in the bit cells to sample the receive data.

Transitions

To work properly, the receive data stream requires transitions. NRZ data encoding does not guarantee transitions in all cases (for example, a long string of zeroes), but the other data encodings do. NRZI guarantees transitions because of inserted zeroes. The Biphase encodings all have at least one transition per bit cell.

Downloaded from www.Manualslib.com manuals search engine

DPLL-tracked bit cell boundaries

The DPLL counter normally counts by 16 but if a transition occurs earlier or later than expected, the count is modified during the next count cycle.

- If the transition occurs earlier than expected, the bit cell boundaries are early with respect to the DPLL-tracked cell boundaries and the count is shortened by either one or two counts.
- If the transition occurs later than expected, the bit cell boundaries are late with respect to the DPLL-tracked bit cell boundaries and the count is lengthened by either one or two counts.

How far off the DPLL-tracked bit cell boundaries are determines whether the count is adjusted by one or two. This tracking allows for minor differences in the transmit and receive clock frequencies.

NRZ and NRZI data encoding

With NRZ and NRZI data encoding, the DPLL counter runs continuously and adjusts after every receive data transition.

Because NRZ encoding does not guarantee a minimum density of transitions, the difference between the sending data rate and the DPLL output clock rate must be very small, and depends on the longest possible run of zeros in the received frame.

NRZI encoding guarantees at least one transition every six bits (with the inserted zeroes). Because the DPLL can adjust by two counts every bit cell, the maximum difference between the sending data rate and the DPLL output clock rate is 1/48 (~2%).

Biphase data encoding

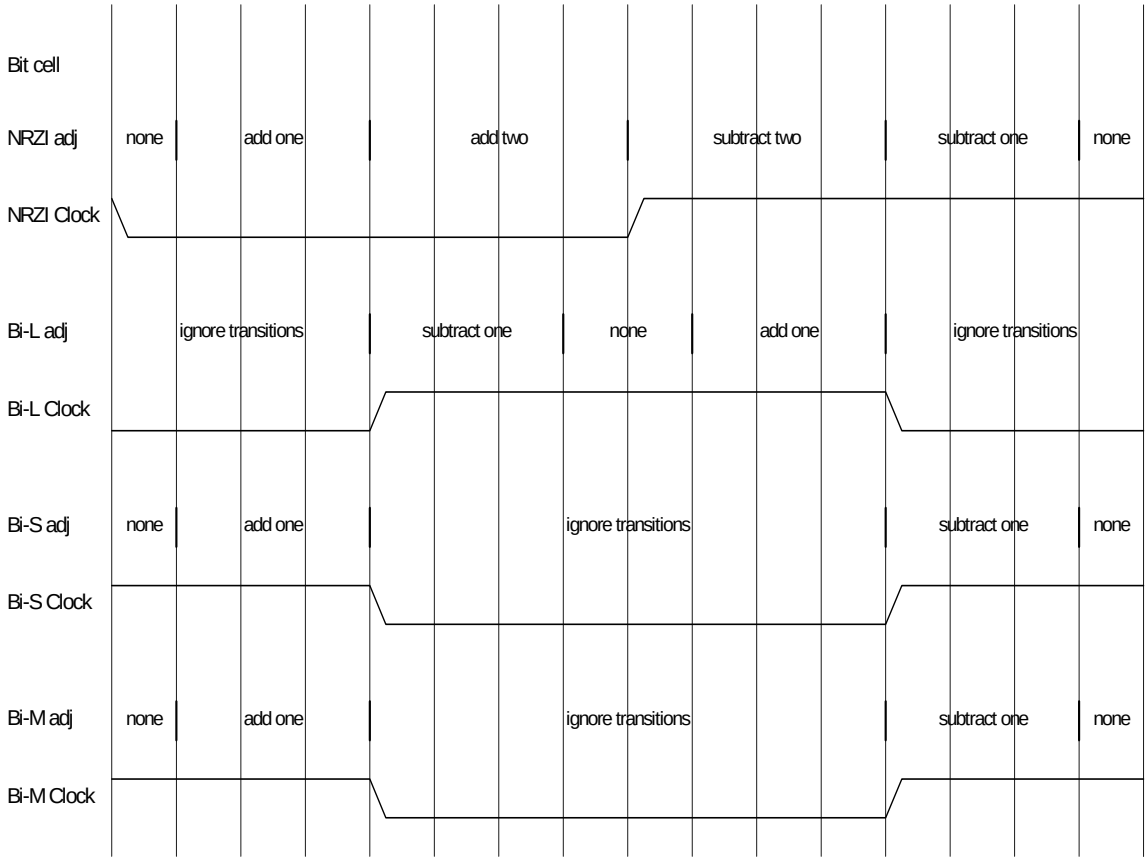
With biphase data encoding, the DPLL works in multiple-access conditions where there may not be flags on the idle line. The DPLL properly generates an output clock based on the first transition in the leading zero of an opening flag. Similarly, the DPLL requires only the completion of the closing flag to provide the extra two clocks to the receiver to properly assemble the data.

- In biphase-level mode, this means the transition that defines the last zero of the closing flag.
- In the biphase-mark and biphase-space modes, this means the transition that defines the end of the last zero of the closing flag.

DPLL operation: Adjustment ranges and output clocks

This figure shows the adjustment ranges and output clock for the different DPLL modes of operation:

Downloaded from www.Manualslib.com manuals search engine



NRZ and NRZI encoding

With NRZ and NRZI encoding, all transitions occur on bit-cell boundaries and the data should be sampled in the middle of the bit cell.

- If a transition occurs after the expected bit-cell boundary, but before the midpoint, the DPLL needs to lengthen the count to line up the bit-cell boundaries; this corresponds to the “add one” and “add two” regions of the figure.
- If a transition occurs before the bit-cell boundary, but after the midpoint, the DPLL needs to shorten the count to line up the bit-cell boundaries; this corresponds to the “subtract one” and “subtract two” regions shown in the figure.
- The DPLL makes no adjustment if the bit-cell boundaries are lined up within one count of the divide-by-sixteen counter. The regions that adjust the count by two allow the DPLL to synchronize faster to the data stream when starting up.

Biphase-Level encoding

With biphase-level encoding, there is a guaranteed “clock” transition at the center of every bit-cell and optional “data” transitions at the bit-cell boundaries. The DPLL

Downloaded from www.Manualslib.com manuals search engine

only uses the clock transitions to track the bit-cell boundaries, by ignoring all transitions occurring outside a window around the center of the bit-cell. The window is half a bit-cell wide.

Because the clock transitions are guaranteed, the DPLL requires that they always be present. If no transition is found in the window around the center of the bit-cell for two successive bit-cells, the DPLL is not in lock and immediately enters search mode. Search mode presumes that the next transition seen is a clock transition and immediately synchronizes to this transition. No clock output is provided to the receiver during the search operation.

Biphase-Mark and Biphase-Space encoding

Biphase-mark and biphase-space encoding are identical per the DPLL and are similar to biphase-level. The primary difference is the clock placement and data transitions. With these encodings, the clock transitions are at the bit-cell boundary and the data transitions are at the center of the bit-cell; the DPLL operation is adjusted accordingly. Decoding biphase-mark or biphase-space encoding requires that the data be sampled by both edges of the recovered receive clock.

IRDA-compliant encode

There is an optional IRDA-compliant encode and decode function available. The encoder sends an active-high pulse for a zero and no pulse for a one. The pulse is 1/4th of a bit-cell wide. The decoder watches for active-low pulses which are stretched to one bit time wide to recreate the normal asynchronous waveform for the receiver. enabling the IRDA-compliant encode/decode modifies the transmitter so there are always two opening flags transmitted.

Normal mode operation

The HDLC achieves normal mode operation by programming the HDLC and Wrapper configuration registers.

Example configuration

This example shows a normal mode operation configuration for a typical application. Any field not specified in this table can be left at reset value.

Control register	Field	Value	Comment
HDLC Control register	CLK	0x3	Enable internal clock generation
HDLC Clock Divider High	EN	0x1	Enable the internal clock divider; the clock rate will be 1.8432 Mbps.

Downloaded from www.Manualslib.com manuals search engine

Control register	Field	Value	Comment
Wrapper Configuration register	RXEN	1	Enable Wrapper receive function
	TXEN	1	Enable Wrapper transmit function

Wrapper and HDLC Control and Status registers

The configuration registers for the HDLC module are located at 0x9002_9000.

Register address map

These are the configuration registers located within a single HDLC module.

Address	Register
9002_9000	Wrapper Configuration
9002_9004	Interrupt Enable
9002_9008	Interrupt Status
9002_9100	HDLC Data Register 1
9002_9104	HDLC Data Register 2
9002_9108	HDLC Data Register 3
9002_910C	Reserved
9002_9110	HDLC Control Register 1
9002_9114	HDLC Control Register 2
9002_9118	HDLC Clock Divider Low
9002_911C	HDLC Clock Divider High

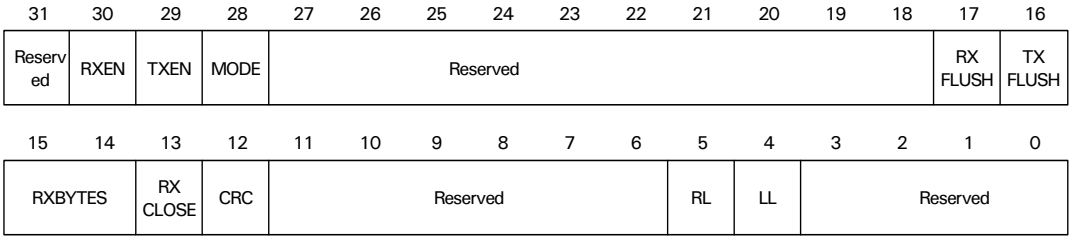
Wrapper Configuration register

Address: 9002_9000

This is the primary Wrapper Configuration register.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31	N/A	Reserved	N/A	N/A
D30	R/W	RXEN	0	0 Disable wrapper function 1 Enable wrapper to process receive characters
D29	R/W	TXEN	0	0 Disable wrapper transmitter function 1 Enable wrapper to process transmit characters
D28	R/W	MODE	0	Applies only to UART channel C. 0 UART mode 1 HDLC mode
D27:18	N/A	Reserved	N/A	N/A
D17	R/W	RXFLUSH	0	Resets the contents of the 64-byte RXFIFO. Write a 1, then a 0 to reset the FIFO.
D16	R/W	TXFLUSH	0	Resets the contents of the 64-byte TX FIFO. Write a 1, then a 0 to reset the FIFO.
D15:14	R	RXBYTES	00	Indicates how many bytes are pending in the wrapper. The wrapper writes to the RX FIFO only when 4 bytes are received or a buffer close event occurs, such as end of frame.
D13	R/W	RXCLOSE	0	Allows software to close a receive buffer. Hardware clears this bit when the buffer has been closed. 0 Idle or buffer already closed 1 Software initiated buffer close
D12	R/W	CRC	0	Controls whether the HDLC transmitter hardware sends CRC bytes before the closing flag. 0 Send CRC bytes before the closing flag 1 Do not send CRC bytes before the closing flag; handled by software
D11:06	N/A	Reserved	0	N/A
D05	R/W	RL	0	**Remote loopback** Provides an internal remote loopback feature. When the RL field is set to 1, the receive HDLC data signal is connected to the transmit HDLC data signal.

Downloaded from www.Manualslib.com manuals search engine

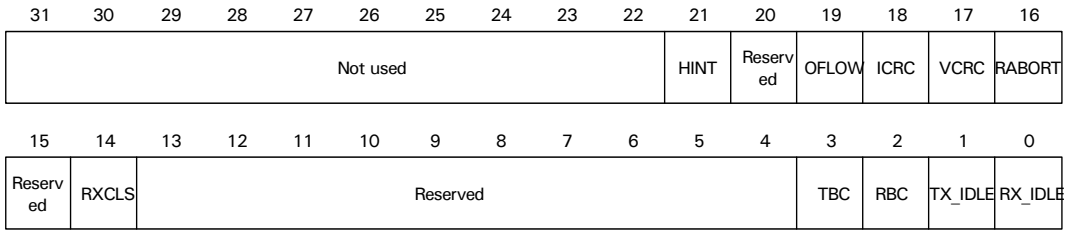
Bits	Access	Mnemonic	Reset	Description
D04	R/W	LL	0	**Local loopback** Provides an internal local loopback feature. When the LL field is set to 1, the transmit HDLC data signal is connected to the receive HDLC data signal.
D03:00	N/A	Reserved	N/A	N/A

Interrupt Enable register

Address: 9002_9004

Use the Interrupt Enable register to enable interrupt generation on specific events. Enable the interrupt by writing a 1 to the appropriate bit field(s).

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:22	R/W	Not used	0	Write this field to 0.
D21	R/W	HINT	0	**Enable HDLC interrupt** Enables interrupt generation directly from the HDLC module. This is normally handled by hardware.
D20	N/A	Reserved	N/A	N/A
D19	R/W	OFLOW	0	**Enable overflow error** Enables interrupt generation if the 4-character FIFO in the HDLC overflows. Note: This should not happen in a properly configured system.
D18	R/W	ICRC	0	**Enable invalid CRC** Enables interrupt generation when a frame is received with an invalid CRC.
D17	R/W	VCRC	0	**Enable valid CRC** Enables interrupt generation when a frame is received with a valid CRC.

Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Reset	Description
D16	R/W	RABORT	0	**Enable receive abort error** Enables interrupt generation when a frame is received with an abort.
D15	N/A	Reserved	N/A	N/A
D14	R/W	RXCLS	0	**Software receive close** Enables interrupt generation when software forces a buffer close.
D13:04	N/A	Reserved	N/A	N/A
D03	R/W	TBC	0	**Enable transmit buffer close** Enables interrupt generation when the HDLC transmit FIFO indicates to the HDLC transmitter that a byte corresponds to a buffer close event.
D02	R/W	RBC	0	**Enable receive buffer close** Enables interrupt generation whenever a buffer close event is passed from the HDLC receiver to the receive FIFO. These are the HDLC receive buffer close events: 1 Receive overrun detected 2 Receive abort detected 3 Buffer closed due to invalid CRC 4 Buffer closed due to valid CRC
D01	R/W	TX_IDLE	0	**Enable transmit idle** Enables interrupt generation whenever the transmitter moves from the active state to the idle state. This indicates that the transmit FIFO is empty and the transmitter is not actively shifting out data.
D00	R/W	RX_IDLE	0	**Enable receive idle** Enables interrupt generation whenever the receiver moves from the active state to the idle state. If a start bit is not received after a stop bit, the receiver enters the idle state.

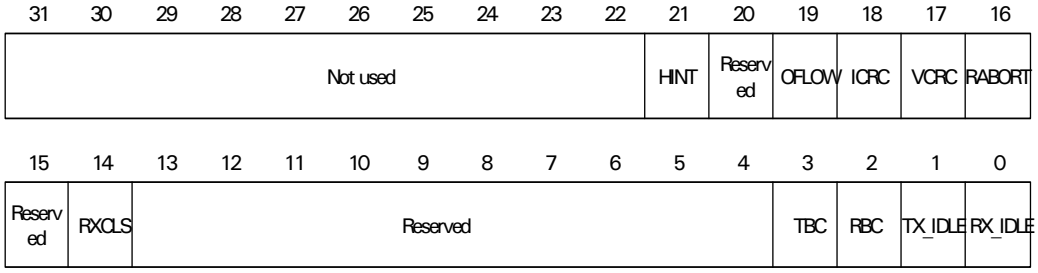
Interrupt Status register

Address: 9002_9008

The Interrupt Status register provides status about HDLC events. All events are indicated by reading a 1 and are cleared by writing a 1.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:22	R/W	Not used	0	Write this field to 0.
D21	R/W1TC	HINT	0	**HDLC interrupt** Indicates that the HDLC has generated an interrupt.
D20	N/A	Reserved	N/A	N/A
D19	R/W1TC	OFLOW	0	**Enable overflow error** Indicates that an overflow occurred in the HDLC's 4-byte FIFO. Note: This should not happen in a properly configured system.
D18	R/W1TC	ICRC	0	**Invalid CRC** Indicates that a frame has been received with a CRC error.
D17	R/W1TC	VCRC	0	**Valid CRC** Indicates that a frame has been received with a valid CRC.
D16	R/W1TC	RABORT	0	**Receive abort error** Indicates that a frame has been received with an abort.
D15	N/A	Reserved	N/A	N/A
D14	R/W1TC	RXCLS	0	**Software receive close** Indicates a software-initiated buffer close has completed.
D13:04	N/A	Reserved	N/A	N/A
D03	R/W1TC	TBC	0	**Transmit buffer close** Indicates that transmission of the last byte in a transmit buffer has completed.
D02	R/W1TC	RBC	0	**Receive buffer close** Indicates that a HDLC receive buffer close condition has occurred. These are HDLC receive buffer close events: 1 Receive overrun detected 2 Receive abort detected 3 Buffer closed due to invalid CRC 4 Buffer closed due to valid CRC

Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Reset	Description
D01	R/W1TC	TX_IDLE	0	**Transmit idle** Indicates that the transmitter has moved from the active state to the idle state. The transmitter moves from the active state to the idle state when the transmit FIFO is empty and the transmitter is not actively shifting out data.
D00	R/W1TC	RX_IDLE	0	**Receive idle** Indicates that the receiver has moved from the active state to the idle state. The receiver moves from the active state to the idle state when a start bit has not been received after the previous stop bit.

HDLC Data Register 1

Address: 9002_9100

HDLC Data Register 1 reads data from the receive buffer and load data in the transmit buffer. This register is for debug purposes only.

Register

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:08	N/A	Reserved	0	N/A
D07:00	R/W	HDATA	0	Read Returns the contents of the receive buffer Write Loads the transmit buffer with a byte of data

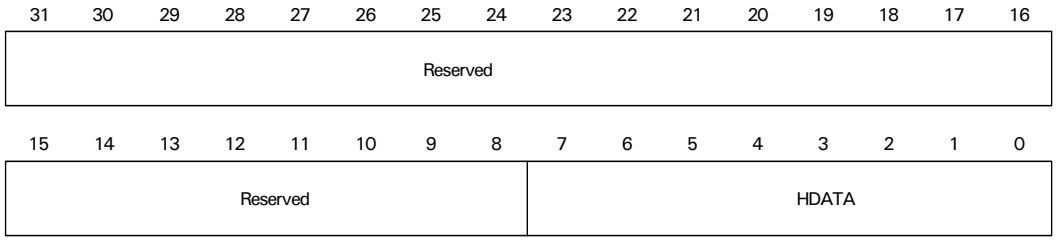
HDLC Data Register 2

Address: 9002_9104

HDLC Data Register 2 writes the last byte of data of a frame after which the CRC and closing flag are transmitted. This register is for debug purposes only.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

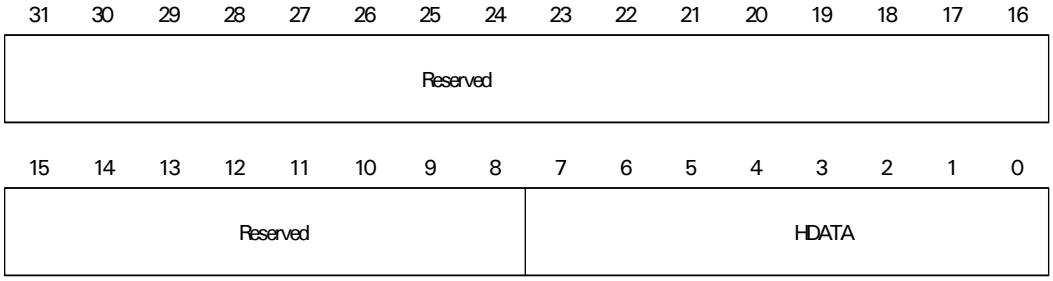
Bits	Access	Mnemonic	Reset	Description	
D31:08	N/A	Reserved	N/A	N/A	
D07:00	R/W	HDATA	0	Read	Returns the contents of the receive buffer
				Write	Used for the last data byte in a frame, after which the CRC and closing flag are transmitted

HDLC Data register 3

Address: 9002_9108

HDLC Data Register 3 writes the last byte of data of a frame after which the closing flag is transmitted. This register is for debug purposes only.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description	
D31:08	N/A	Reserved	N/A	N/A	
D07:00	R/W	HDATA	0	Read	Returns the contents of the receive buffer
				Write	Used for the last data byte in a frame, after which the closing flag is transmitted

Downloaded from www.Manualslib.com manuals search engine

HDLC Control Register 1

Address: 9002_9110

HDLC Control Register 1 configures the HDLC transmitter and receiver.

Register

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:08	R	Not used	0	Write this field to 0.
D07:06	R	HMODE	0	00 Normal operation 01 Force receiver to flag search mode 10 Normal operation 11 Force transmitter to send abort
D05:04	N/A	Reserved	N/A	N/A
D03:02	R/W	CLK	0	**Clock source** Note: This field should be programmed last 00 Reserved 01 Reserved 10 Use external clock 11 Use internal clock
D01	R/W	Not used	0	Always write 0 to this bit.
D00	R/W	HINT	0	0 Disable the HDLC interrupt 1 Enable the HDLC interrupt

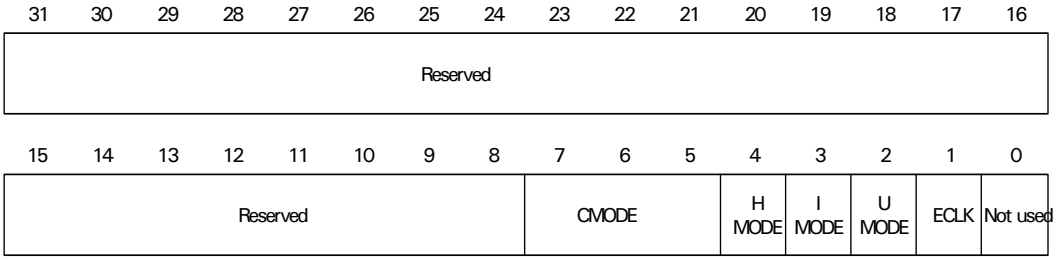
HDLC Control Register 2

Address: 9002_9114

HDLC Control Register 2 configures the HDLC transmitter and receiver.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:08	R	Not used	0	Write this field to 0.
D07:05	R/W	CMODE	0	**Coding mode** 000 NRZ data encoding for receiver and transmitter 010 RZI data encoding for receiver and transmitter 100 Biphase-Level (Manchester) data encoding for receiver and transmitter 110 Biphase-Space data encoding for receiver and transmitter 111 Biphase-Mark data encoding for receiver and transmitter
D04	R/W	HMODE	0	**HDLC mode** 0 Normal HDLC data encoding 1 Enable NRZI coding (1/4 bit-cell IRDA-compliant). This mode can be used only with internal clock and NRZ data encoding.
D03	R/W	IMODE	0	**Transmit idle mode** 0 Transmit flags while in idle mode 1 Transmit all 1s while in idle mode
D02	R/W	UMODE	0	**Underrun mode** 0 Transmit flag on underrun 1 Transmit abort on underrun
D01	R/W	ECLK	0	**External clock mode** 0 The HDLC module will use separate external receive and transmit clocks 1 The HDLC receiver and transmitter will both use the external transmit clock.
D00	R	Not used	0	Always write 0 to this bit.

HDLC Clock Divider Low

Address: 9002_9118

Downloaded from www.Manualslib.com manuals search engine

Use the HDLC CLock Divider Low register to set bits 07:00 of the clock divider. This is the equation for the HDLC clock rate:

$$\text{HDLC rate (bps)} = \frac{29.4912\ \text{MHz}}{16 \times (\text{DIV} = 1)}$$

Register

Register bit assignment

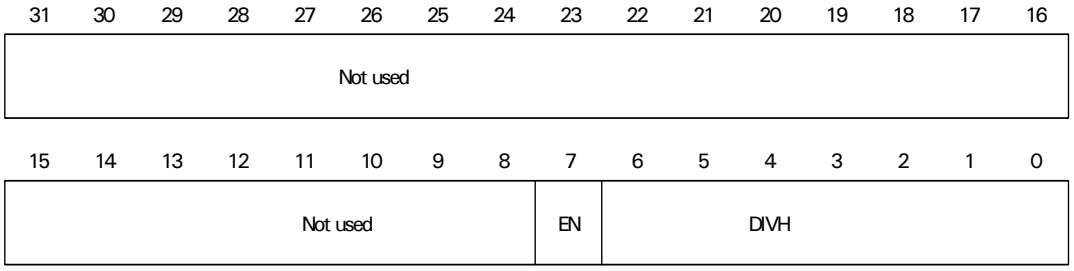
Bits	Access	Mnemonic	Reset	Description
D31:08	R	Not used	0	Write this field to 0.
D07:00	R/W	DIVL	0	Eight LSBs of the divider that generates the HDLC transmit and receive clock.

HDLC Clock Divider High

Address: 9002_911C

Use the HDLC CLock Divider High register to set bits 14:08 of the clock divider.

Register



Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:08	R	Not used	0	Write this field to 0.
D07	R/W	EN	0	**Clock enable** Must be set when the internal clock is used.
D06:00	R/W	DIVH	0	Seven MSBs of the divider that generates the HDLC transmit and receive clock.

Downloaded from www.Manualslib.com manuals search engine

Serial Control Module: SPI

CHAPTER 12

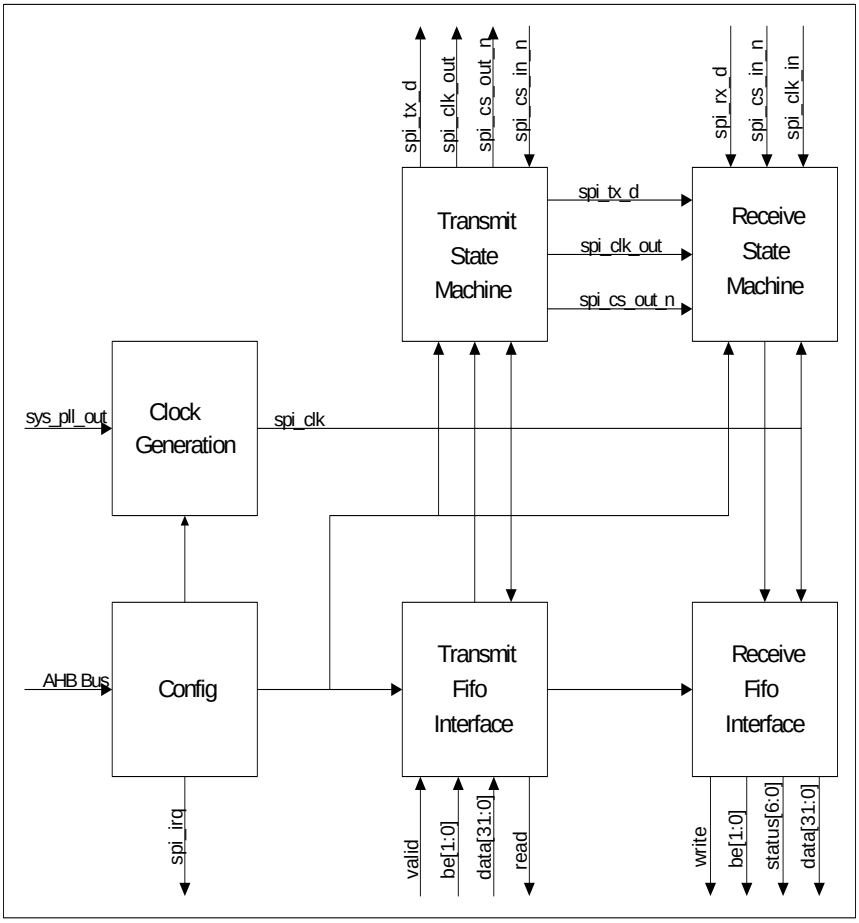
The processor ASIC contains a single high speed, four-wire, serial peripheral interface (SPI) module.

Features

- DMA transfers to and from system memory
- Four-wire interface (RXD, TXD, CLK, CS)
- Multi-drop supported through GPIO programming
- Master or slave operation
- High speed data transfer
  - Master: 33.33 Mbps
  - Slave: 7.50 Mbps
- Programmable MSB/LSB formatting
- Programmable SPI mode (0, 1, 2, or 3)
- Master mode internal diagnostic loopback
- Maskable interrupt conditions
  - Receiver idle
  - Transmitter idle

Downloaded from www.Manualslib.com manuals search engine

SPI module structure



SPI controller

The SPI controller provides a full-duplex, synchronous, character-oriented data channel between master and slave devices, using a four-wire interface (RXD, TXD, CLK, CS#). The master interface operates in a broadcast mode. The slave interface is activated using the CS# signal. You can configure the master interface to address various slave interfaces using the GPIO pins.

Simple parallel/serial data conversion

SPI provides simple parallel/serial data conversion to stream serial data between memory and a peripheral. The SPI port has no protocol associated with it other than transferring information in multiples of 8 bits.

Full duplex operation

The SPI port can operate in full-duplex mode. Information transfer is controlled by a single clock signal. The clock and chip select signals are chip outputs for a master mode operation and inputs for a slave mode operation.

Downloaded from www.Manualslib.com manuals search engine

SPI clocking modes

There are four SPI clocking modes. Each mode's characteristics are defined by the idle value of the clock, which clock edge captures data, and which clock edge drives data. The MODE field in the SPI Configuration register specifies the timing mode.

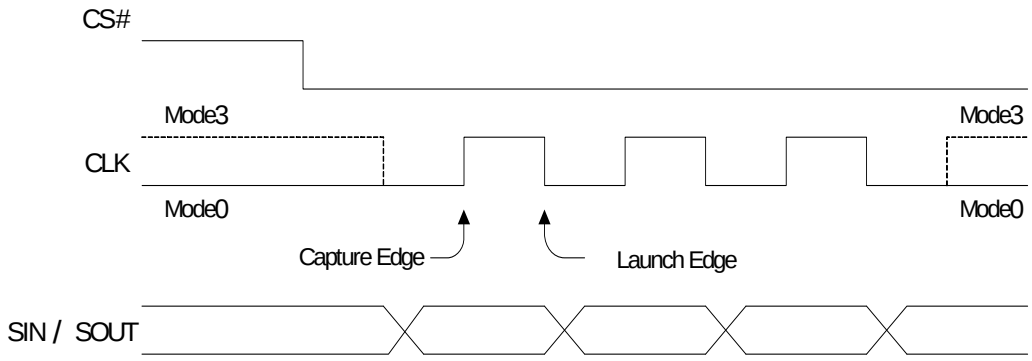
Timing modes

SPI mode	SPI CLK Idle	SPI DATA IN capture edge	SPI DATA OUT drive edge
0	Low	Rising	Falling
1	High	Falling	Rising
2	Low	Falling	Rising
3	High	Rising	Falling

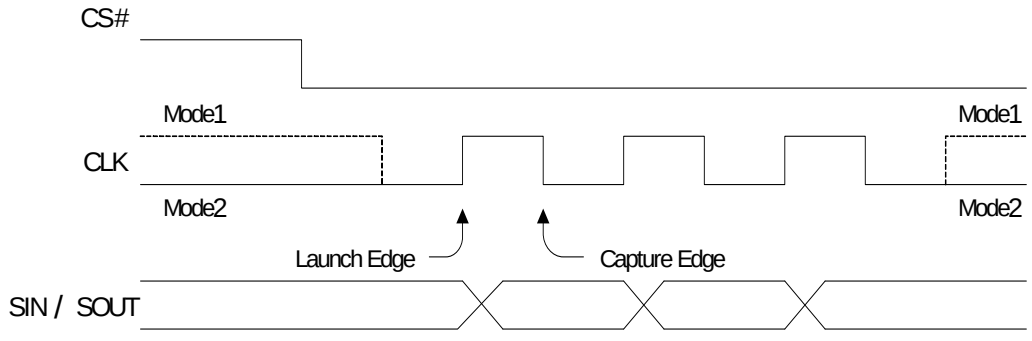
Clocking mode diagrams

The next two diagrams show the four SPI clocking modes. SPI Mode0 and SPI Mode3 are the most commonly used modes.

SPI Mode0 and Mode3 functional timing



SPI Mode1 and Mode2 functional timing



Downloaded from www.Manualslib.com manuals search engine

SPI clock generation

The reference clock for the SPI module is the system PLL output. This clock is a nominal 300 MHz.

- In SPI master mode, the clock is divided down to produce the required data rate.
- In SPI slave mode, the divided down clock recovers the input SPI clock.

Clock generation samples

SPI clock generation is specified using the Clock Generation register. These are some examples of clock generation:

Interface Type	Data rate	DIVISOR
Master	33 Mbps	0x009
Master	20 Mbps	0x00F
Master	5 Mbps	0x03C
Master	500 Kbps	0x258
Slave	all	0x006

In SPI master mode

In SPI master mode, the value programmed in the DIVISOR field must always be rounded up to the next whole integer. For example, if the required data rate is 14 Mbps, the calculation is (300 / 14) or 21.43.

- The value programmed in the DIVISOR field would be 0x016.
- The actual data rate would be 13.64 Mbps.
- The general equation is:

  DIVISOR = round Up (PLL output / interface data rate)

In SPI slave mode

In SPI slave mode, the value programmed in the DIVISOR field should always be 0x006. The SPI slave mode data rate is determined by the frequency of the input clock provided by the external SPI master.

System boot-over-SPI operation

The NET+SPI ASIC boots from an external, non-volatile, serial memory device. The device can be either a serial EEPROM or a serial Flash. In either case, the device must support a four-wire, mode0-compatible SPI interface.

The boot-over-SPI hardware interfaces to devices requiring an 8-bit address, 16-bit address, or 24-bit address. The address width is indicated by strapping pins boot_mode[1:0].

Downloaded from www.Manualslib.com manuals search engine

Available strapping options

boot_mode[1:0]	Address width
00	Disabled
01	8-bit address
10	16-bit address
11	24-bit address

EEPROM/FLASH header

The boot-over-SPI hardware requires several pieces of user-supplied information to complete the boot operation. This information must be located in a 128-byte header starting at address zero in the external memory device. Each entry in the header is four bytes long.

Header format

This is the format of the 128-byte header.

Entry	Name	Description
0x0	Size[19:0] (31:20 reserved)	Total number of words to fetch from the SPI-EEPROM. The total must include the 32-word header: (Code image size in bytes + 128) / 4)
0x4	Mode[27:0] (31:28 reserved)	All SDRAM components contain a Mode register. This register contains control information required to successfully access the component. The fields (available in any SDRAM specification) are defined as follows: ▪Burst length: 4 for 32-bit data bus, 8 for 16-bit data bus ▪Burst type: Sequential ▪CAS latency: Component-specific; 2 or 3 ▪OpMode: Standard ▪Write burst mode: Programmed burst length This value must be left-shifted such that it is aligned to the row address bits as specified in “Address mapping,” beginning on page 229. For example, 4Mx16 components can be combined to create a 32-bit bus. These parts require 12 row address bits. With a CAS2 access, the Mode register contents would be 0x22. This value is shifted 12 places to the left (0x00022000) to form the value in the SDRAM config field.
0x8	Divisor[9:0] (31:10 reserved)	Defines the interface data rate for the boot-over-SPI operation after the initial 16-bytes. A data rate of about 375 Kbps fetches the 16-byte header. See the Clock Generation register for more details.
0xc	HS Read[0] (31:1 reserved)	A 1 indicates the external device supports high-speed read operation. Serial FLASH devices operating above 20MHz generally support this feature.
0x10	Config register	See the Memory Controller chapter.

Downloaded from www.Manualslib.com manuals search engine

Entry	Name	Description
0x14	DynamicRefresh	See the Memory Controller chapter. For example, the value of this entry is 0x00000025 given a 74.9 MHz AHB clock and a 7.8125µs refresh period.
Ox18	DynamicReadConfig	See the Memory Controller chapter.
0x1c	DynamictRP	See the Memory Controller chapter
Ox20	DynamictRAS	See the Memory Controller chapter
Ox24	DynamictSREX	See the Memory Controller chapter
Ox28	DynamictAPR	See the Memory Controller chapter
Ox2c	DynamictDAL	See the Memory Controller chapter
Ox30	DynamictWR	See the Memory Controller chapter
Ox34	DynamictRC	See the Memory Controller chapter
Ox38	DynamictRFC	See the Memory Controller chapter
Ox3c	DynamictXSRt	See the Memory Controller chapter
Ox40	DynamictRRD	See the Memory Controller chapter
Ox44	DynamictMRD	See the Memory Controller chapter
Ox48	DynamictConfig0	See the Memory Controller chapter. Field B (buffer enable, in the DynamicConfig0 register) should be set to 0 (buffers disabled). The buffers will be enabled by hardware as part of the boot process.
Ox4c	DynamictRasCas0	See the Memory Controller Chapter
Ox50- Ox7c	Reserved	
Ox80	Boot Code	First 4 bytes of boot code

Time to completion

The boot-over-SPI operation is performed in two steps.

- In the first step, the hardware fetches the 16-byte header. The data rate for this step is about 375 Kbps and completes in less than 0.5ms.
- In the second step, the hardware fetches the image at the user-specified data rate. Calculate time to completion for this step as shown:

  $$Time(s) = (1 / data\_rate) * IMAGE_{SIZE}$$

  For example, with a 20 Mbps data rate and a 256 KB (2Mb) image, the time to completion is approximately 105ms.

Downloaded from www.Manualslib.com manuals search engine

SPI Control and Status registers

The configuration registers for the SPI module are located at 0x9003_1000.

Register address map

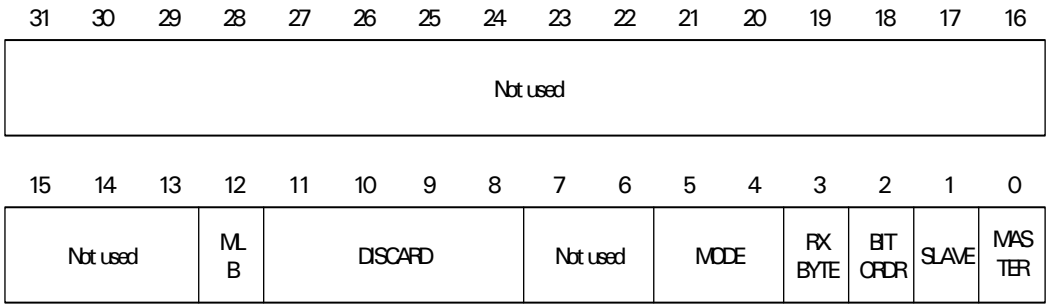
Address	Register
9003_1000	SPI Configuration register
9003_1010	Clock Generation register
9003_1020	Interrupt Enable register
9003_1024	Interrupt Status register

SPI Configuration register

Address: 9003_1000

This is the primary SPI Configuration register.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:13	R/W	Not used	0	Write this field to 0.
D12	R/W	MLB	0	**Enable master loopback mode** Write a 1 to enable the master mode transmitter to receiver loopback function.

Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Reset	Description
D11:08	R/W	DISCARD	0	**Discard bytes** Defines the number of bytes the receiver should drop when the transmitter has initiated a new operation. ■ A new operation is defined by the chip select signal being asserted low. ■ The programmed value defines the number of bytes to discard. ■ The maximum number of receive bytes that can be discarded is 14.
D07:06	R/W	Not used	0	Write this field to 0.
D05:04	R/W	MODE	0	**SPI mode** Defines the required interface timing as specified in “Timing modes” on page 435.
D03	R/W	RXBYTE	0	Controls how the SPI receiver handles receive data. ■ RXBYTE set to 0 — The receiver buffers 4 bytes before writing to the RX FIFO. ■ Write a 1 to RXBYTE — The receiver writes to the RX FIFO each time a new byte is received. This allows low latency handling of SPI receive data.
D02	R/W	BITORDR	0x0	**Bit ordering** Controls the order in which bits are transmitted and received in the serial shift register. ■ BITORDR set to 0 — Bits are processed LSB first, MSB last. ■ BITORDR set to 1 — Bits are processed MSB first, LSB last.
D01	R/W	SLAVE	0	**Slave enable** Set this field to 1 to enable the SPI module for slave operation. The SLAVE field must not be set until all SPI configuration fields have been defined. You can set either the MASTER field (D00) or the SLAVE field, but not both.
D00	R/W	MASTER	0	**Slave enable** Set this field to 1 to enable the SPI module for master operation. The MASTER field must not be set until all SPI configuration fields have been defined. You can set either the MASTER field or the SLAVE field (D01), but not both.

Clock Generation register

Address: 9003_1010

Downloaded from www.Manualslib.com manuals search engine

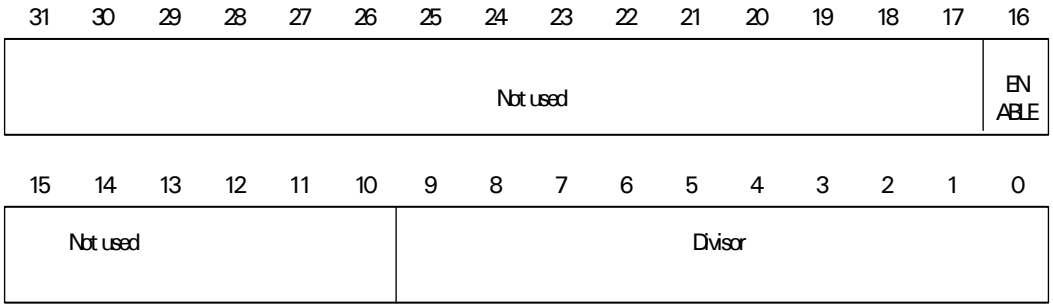
Use this register to define the data rate of the interface.

This register must be programmed in three steps. Failure to follow these steps can result in unpredictable behavior of the SPI module.

Register programming steps

1. Set the ENABLE field to 0. The DIVISOR field must not be changed.
2. Set the DIVISOR field to the value you want.
3. Set the ENABLE field to 1. The DIVISOR field must not be changed.

Register



Register bit assignment

Bit(s)	Access	Mnemonic	Reset	Divisor
D31:17	R/W	Not used	0	Write this field to 0.
D16	R/W	ENABLE	0	**Enable clock generation** Write a 1 to this field to enable the SPI module clock generation logic.
D15:10	R/W	Not used	0	Write this field to 0.
D09:00	R/W	DIVISOR	0	**Divisor** Allows you to specify the required data rate of the interface. The reference clock used is the system PLL output. This frequency is a nominal 300 MHz. ■ For SPI master operation — Set this field to a value no smaller than 0x009. This produces the maximum supported data rate of 33 Mbps. ■ For SPI slave operation — Always set this field to 0x006.

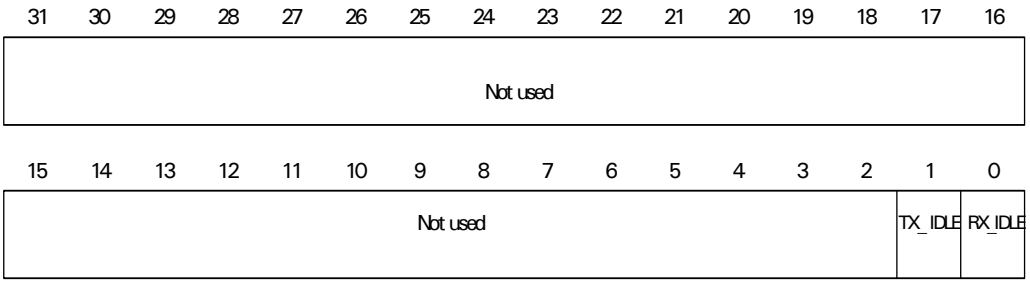
Interrupt Enable register

Address: 9003_1020

Downloaded from www.Manualslib.com manuals search engine

Use the Interrupt Enable register to enable interrupt generation on specific events. Enable the interrupt by writing a 1 to the appropriate bit field(s).

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:02	R/W	Not used	0	Write this field to 0.
D01	R/W	TX_IDLE	0	**Enable transmit idle** Enables interrupt generation whenever the transmitter moves from the active state to the idle state. ■ In master mode, this indicates that the transmit FIFO is empty and that the transmitter is not actively shifting out data. ■ In slave mode, this indicates that the externally provided chip select has been deasserted.
D00	R/W	RX_IDLE	0	**Enable receive idle** Enables interrupt generation whenever the receiver moves from the active state to the idle state. In either master or slave mode, this indicates that the chip select signal has been deasserted.

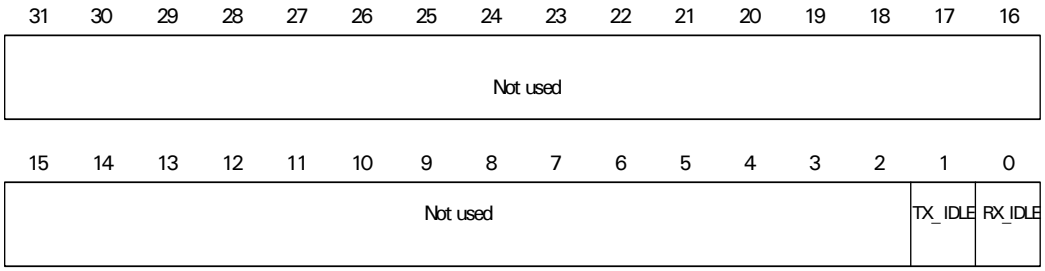
Interrupt Status register

Address: 9003_1024

The Interrupt Status register provides status about SPI events. All events are indicated by reading a 1 and are cleared by writing a 1.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:02	R/W	Not used	0	Write this field to 0.
D01	R/W1TC	TX_IDLE	0	**Transmit idle** Indicates that the transmitter has moved from the active state to the idle state. The transmitter moves from the active state to the idle state when the transmit FIFO is empty and the transmitter is not actively shifting out data.
D00	R/W1TC	RX_IDLE	0	**Receive idle** Indicates that the receiver has moved from the active state to the idle state. The receiver moves from the active state to the idle state when a start bit has not been received within 4 bit periods of the previous stop bit.

SPI timing characteristics

These are the guaranteed timing parameters for all four SPI clocking modes.

SPI master timing parameters

Parm	Description	Min	Max	Unit	Notes
S1	CS# falling to CLK rising	1		clock	1
S2	CLK period low time	12	13	ns	2
S3	CLK period high time	12	13	ns	2
S4	Data output setup to CLK rising	11		ns	3
S5	Data output hold from CLK rising	11		ns	3
S6	Data input setup to CLK rising	10		ns	4
S7	Data input hold from CLK rising	0		ns	4
S8	CLK falling to CS# rising	1		clock	1
S9	CS# deassertion time	4		clock	1

Downloaded from www.Manualslib.com manuals search engine

Notes:

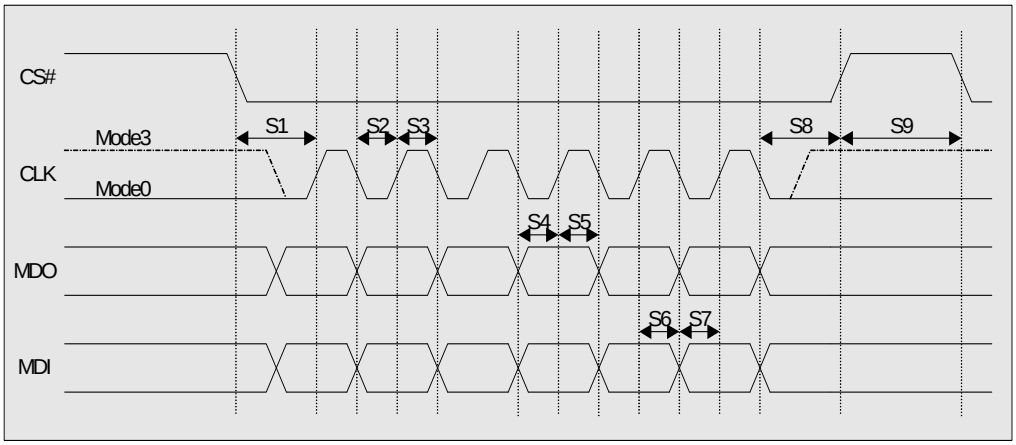
1 The unit *clock* refers to the SPI master clock.

2 The SPI master interface clock duty cycle is always at least 52/48. The numbers shown here are for a 40 Mhz clock rate.

3 The numbers shown here are for a 40 Mhz clock rate. Usually, this parameter is one half the SPI master interface clock period less 1.5ns.

4 This parameter does not depend on the SPI master interface clock rate.

SPI master timing diagram



SPI slave timing parameters

Parm	Description	Min	Max	Unit	Notes
S11	CS# falling to CLK rising	50		ns	3
S12	CLK period low time	53	80	ns	1,2
S13	CLK period high time	53	80	ns	1,2
S14	Data input setup to CLK rising	10		ns	4
S15	Data input hold from CLK rising	15		ns	3
S16	Data output setup to CLK rising	80		ns	2
S17	Data output hold from CLK rising	67		ns	2
S18	CLK falling to CS# rising	50		ns	3
S19	CS# deassertion time	266		ns	2

Notes:

1 The SPI slave interface clock duty cycle should be no worse than 60/40.

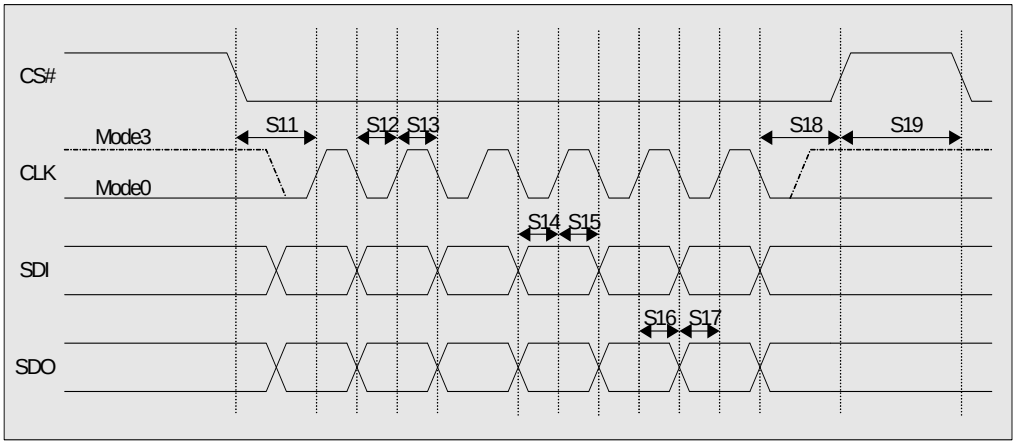
Downloaded from www.Manualslib.com manuals search engine

2 The numbers shown here are for a 7.5 Mhz SPI slave interface clock rate.

3 The numbers shown here are for a 300 Mhz PLL output frequency. This value must be proportionally increased with a PLL output frequency decrease.

4 This parameter does not depend on any clock frequency.

SPI slave timing diagram



Downloaded from www.Manualslib.com manuals search engine

Downloaded from www.Manualslib.com manuals search engine

I2C Master/Slave Interface

CHAPTER 13

The I2C master/slave interface provides an interface between the ARM CPU and the I2C bus.

The I2C master/slave interface basically is a parallel-to-serial and serial-to-parallel converter. The parallel data received from the ARM CPU has to be converted to an appropriate serial form to be transmitted to an external component using the I2C bus. Similarly, the serial data received from the I2C bus has to be converted to an appropriate parallel form for the ARM CPU. The I2C master interface also manages the interface timing, data structure, and error handling.

Overview

The I2C module is designed to be a master and slave. The slave is active only when the module is being addressed during an I2C bus transfer; the master can arbitrate for and access the I2C bus only when the bus is free (idle) — therefore, the master and slave are mutually exclusive.

Physical I^2C bus

The physical I^2C bus consists of two open-drain signal lines: serial data (SDA) and serial clock (SCL). Pullup resistors are required; see the standard I^2C bus specification for the correct value for the application. Each device connected to the bus is software-addressable by a unique 7- or 10-bit address, and a simple master/slave relationship exists at all times.

A master can operate as a master-transmitter (writes)) or a master-receiver (reads). The slaves respond to the received commands accordingly:

- In transmit mode (slave is read), the host interface receives character-based parallel data from the ARM. The module converts the parallel data to serial format and transmits the serial data to the I^2C bus.
- In receive mode (slave is written to), the I^2C bus interface receives 8-bit-based serial data from the I^2C bus. The module converts the serial data to parallel format and interrupts the host. The host's interrupt service routine reads the parallel data from the data register inside the I^2C module. The serial data stream synchronization and throttling are done by modulating the

Downloaded from www.Manualslib.com manuals search engine

serial clock. Serial clock modulation can be controlled by both the transmitter and receiver, based in their hosts' service speed.

Multi-master bus

The I^2C is a true multi-master bus with collision detection and arbitration to prevent data corruption when two or more masters initiate transfer simultaneously. If a master loses arbitration during the addressing stage, it is possible that the winning master is trying to address the transfer. The losing master must therefore immediately switch over to its slave mode.

The on-chip filtering rejects spikes on the bus data line to preserve data integrity. The number of ICs that can be connected to the same bus is limited only by a maximum bus capacity of 400 pf.

I^2C external addresses

I^2C external [bus] addresses are allocated as two groups of eight addresses (0000XXX and 1111XXX)

:

Slave addres	R/W bit	Description
0000 000	0	General call address
0000 000	1	START byte (not supported in the processor)
0000 001	X	CBUS address (not supported in the processor)
0000 010	X	Reserved for different bus format
0000 011	X	Reserved
0000 1xx	X	hs-mode master code (not supported in the processor)
1111 1xx	X	Reserved
1111 0xx	X	10-bit slave address

The general call address is for addressing all devices connected to the I^2C bus. A device can ignore this address by not issuing an acknowledgement. The meaning of the general call address is always specified in the second byte.

Downloaded from www.Manualslib.com manuals search engine

I²C command interface

The I²C module converts parallel (8-bit) data to serial data and serial data to parallel data between the processor and the I²C bus, using a set of interface registers.

- The primary interface register for transmitting data is the CMD_TX_DATA_REG (write-only).
- The primary interface register for receiving data is the STATUS_RX_DATA_REG (read-only).

Locked interrupt driven mode

I²C operates in a *locked interrupt driven mode*, which means that each command issued must wait for an interrupt response before the next command can be issued (illustrated in "Flow charts," beginning on page 457).

The first bit of the command — 0 or 1 — indicates to which module — master or slave, respectively — the command in the CMD field (of the CMD_TX_DATA_REG) is sent. The master module can be sent a master command only; the slave module can be sent a slave command only (see "Master module and slave module commands," beginning on page 449, for a list of commands). If a command is sent to the master module, that module is locked until a command acknowledgement is given. Similarly, if a command is sent to the slave module, the slave module is locked until it receives a command acknowledgement. With either module, the acknowledgement can be any interrupt associated with that module. When a module is locked, another command must not be sent to that module.

The command lock status can be checked in the STATUS_RX_DATA_REG.

Master module and slave module commands

The I²C master recognizes four high-level commands, which are used in the CMD field of the Command register; the I²C slave recognizes two high-level commands:

Command	Name	Description
0x0	M_NOP	No operation.
0x4	M_READ	Start reading bytes from slave.
0x5	M_WRITE	Start writing bytes to slave.
0x6	M_STOP	Stop this transaction (give up the I²C bus).
0x10	S_NOP	No operation. This command is necessary for 16-bit mode, providing data in TX_DATA_REG without a command.
0x16	S_STOP	Stop transaction by not acknowledging the byte received.

Bus arbitration

Any M_READ or M_WRITE command causes the I²C module to participate in the bus arbitration process when the I²C bus is free (idle). If the module becomes the new

Downloaded from www.Manualslib.com manuals search engine

bus owner, the transaction goes through. If the module loses bus arbitration, an M_ARBIT_LOST interrupt is generated to the host processor and the command must be reissued.

I²C registers

All registers have 8-bit definitions, but must be accessed in pairs. For example, TX_DATA_REG and CMD_REG are written simultaneously and RX_DATA_REG and STATUS_REG are read simultaneously.

Register address map

This table shows the register addresses. All configuration registers must be accessed as 32-bit words and as single accesses only. Bursting is not allowed.

Register	Description
9005 0000	Command Transmit Data register (CMD_TX_DATA_REG) Status Receive Data register (STATUS_RX_DATA_REG)
9005 0004	Master Address register
9005 0008	Slave Address register
9005 000C	Configuration register

After a reset, all registers are set to the initial value. If an unspecified register or bit is read, a zero is returned.

Command Transmit Data register

Address: 9005 0000

The Command Transmit Data (CMD_TX_DATA_REG) register is the primary interface register for transmission of data between the I/O hub and I²C bus. This register is write only.

Register

Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

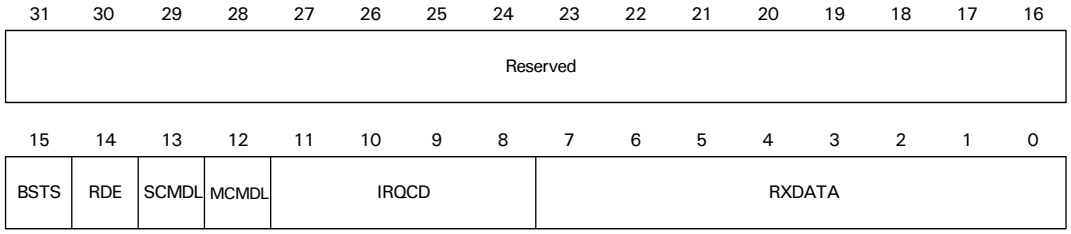
Bits	Access	Mnemonic	Reset	Description
D31:16	N/A	Reserved	N/A	N/A
D15	W	PIPE	0x0	Pipeline mode Must be set to 0.
D14	W	DLEN	0x0	**I^{2}C DLEN port** (iic_dlen) Must be set to 0.
D13	W	TXVAL	0x0	**Provide new transmit data in** CMD_TX_DATA_REG (tx_data_val).
D12:08	W	CMD	0x0	Command to be sent (see "Master module and slave module commands," beginning on page 449)
D07:00	W	TXDATA	0x0	Transmit data to I^{2}C bus.

Status Receive Data register

Address: 9005 0000

The Status Receive Data register (STATUS_RX_DATA_REG) is the primary interface register for receipt of data between the I/O hub and I^{2}C bus. This register is read only.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:16	N/A	Reserved	N/A	N/A
D15	R	BSTS	N/A	Bus status (master only) 0Bus is free 1Bus is occupied
D14	R	RDE	N/A	Receive data enable (rx_data_en) Received data is available.

Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Reset	Description
D13	R	SCMDL	N/A	Slave command lock The Slave Command register is locked.
D12	R	MCMDL	N/A	Master command lock The Master Command register is locked.
D11:08	R	IRQCD	N/A	Interrupt codes (irq_code) The interrupt is cleared if this register is read. See "Interrupt Codes" on page 455 for more information.
D07:00	R	RXDATA	N/A	Received data from I^2C bus Together with a RX_DATA interrupt, this register provides a received byte (see "Master/slave interrupt codes" on page 455).

Master Address register

Address: 9005 0004

If using 7-bit addressing, the master device address field uses only bits D07:01; otherwise, all 10 bits are used.

Register

Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D10:01	R/W	MDA	0x0	Master device address Used for selecting a slave. Represents bits 6:0 of the device address if using 7-bit address. D10:08 are not used. Represents bits 9:0 of device address if using 10-bit address.
D00	R/W	MAM	0x0	Master addressing mode 07 bit address mode 110 bit address mode

Slave Address register

Address: 9005 0008

If using 7-bit addressing, the slave device address field uses only bits D07:01; otherwise, bits 10:01 are used.

Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16
Reserved
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Reserved
Gnrl call addr
Slave device address
Slave addr mode

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D11	R/W	GCA	0x0	General call address (s_gca_irq_en) Enable the general call address.
D10:01	R/W	SDA	0x3FF	Slave device address Represents bits 6:0 of device address if using 7-bit address; D10:08 are not used. Represents bits 9:0 of device address if using 10-bit address.
D00	R/W	SAM	0x0	Slave addressing mode 07 bit address mode 110 bit address mode

Downloaded from www.Manualslib.com manuals search engine

Configuration register

Address: 9005 000C

The Configuration register controls the timing on the I^2C bus. This register also controls the external interrupt indication, which can be disabled.

The I^2C bus clock timing is programmable by the scl_ref value (D08:00). The timing parameter for standard mode is as follows:

$I^2C\_bus\_clock = clk / ((CLREF*2) + 4 + scl\_delay)$

clk = PLL Clk Out/4

Notes: To determine the "PLL Clk Out" frequency, see the "PLL configuration and control system block diagram" on page 152 and the "PLL Configuration register" on page 186. In noisy environments and fast-mode transmission, spike filtering can be applied to the received I^2C data and clock signal. The spike filter evaluates the incoming signal and suppresses spikes. The maximum length of the suppressed spikes can be specified in the spike filter width field of the Configuration register.

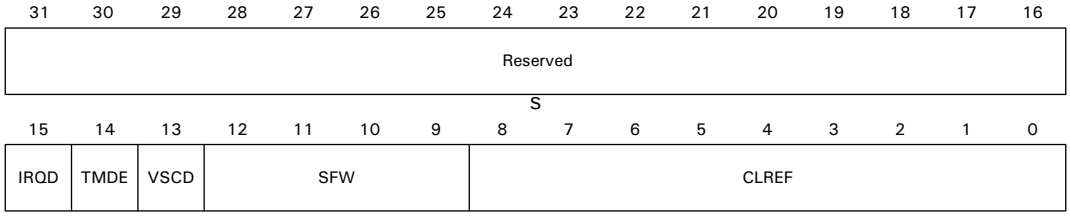
Timing parameter for fast-mode

This is the timing parameter for fast-mode:

$I^2C\_bus\_clock = (4 / 3) \times (clk / ((CLREF*2) + 4 + scl\_delay))$

scl_delay is influenced by the SCL rise time.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:16	N/A	Reserved	N/A	N/A
D15	R/W	IRQD	0	**Mask the interrupt to the ARM CPU** (irq_dis) Must be set to 0.
D14	R/W	TMDE	1	Timing characteristics of serial data and serial clock 0Standard mode 1Fast mode
D13	R/W	VSCD	1	Virtual system clock divider for master and slave Must be set to 0.

Downloaded from www.Manualslib.com manuals search engine

Bits	Access	Mnemonic	Reset	Description
D12:09	R/W	SFW	0xF	Spike filter width A default value of 1 is recommended. Available values are 0–15.
D08:00	R/W	CLREF	0x0	clk_ref[9:1] The I2C clock on port iic_scl_out is generated by the system clock divided by the 10-bit value of clk_ref. The LSB of clk_ref cannot be programmed, and is set to 0 internally. The programmed value of clk_ref[9:1] must be greater than 3.

Interrupt Codes

Interrupts are signaled in the irq_code field in the STATUS_REG, by providing the appropriate interrupt code (see "Master/slave interrupt codes" on page 455). The ARM CPU waits for an interrupt by polling the STATUS_REG or checking the irq signal. An interrupt is cleared by reading the STATUS_REG, which also forces the irq signal down (minimum one cycle if another interrupt is stored).

Note: RX_DATA_REG contains only a received byte if it is accessed after a RX_DATA master or slave interrupt is signaled. At all other times, the internal master or slave shift register is accessed with RX_DATA_REG (see "Status Receive Data register" on page 451).

Master/slave interrupt codes

Code	Name	Master/slave	Description
0x0	NO_IRQ	N/A	No interrupt active
0x1	M_ARBIT_LOST	Master	Arbitration lost; the transfer has to be repeated
0x2	M_NO_ACK	Master	No acknowledge by slave
0x3	M_TX_DATA	Master	TX data required in register TX_DATA
0x4	M_RX_DATA	Master	RX data available in register RX_DATA
0x5	M_CMD_ACK	Master	Command acknowledge interrupt
0x6	N/A	N/A	Reserved
0x7	N/A	N/A	Reserved
0x8	S_RX_ABORT	Slave	The transaction is aborted by the master before the slave performs a NO_ACK.
0x9	S_CMD_REQ	Slave	Command request
$0xA_x$	S_NO_ACK	Slave	No acknowledge by master (TX_DATA_REG is reset)

Downloaded from www.Manualslib.com manuals search engine

Code	Name	Master/slave	Description
0xB	S_TX_DATA_1ST	Slave	TX data required in register TX_DATA, first byte of transaction
0xC	S_RX_DATA_1ST	Slave	RX data available in register RX_DATA, first byte of transaction
0XD	S_TX_DATA	Slave	TX data required in register TX_DATA
0xE	S_RX_DATA	Slave	RX data available in register RX_DATA
0XF	S_GCA	Slave	General call address

Software driver

I²C master software driver

The I²C master software driver uses three commands only:

- M_READ to start a read sequence
- M_WRITE to start a write sequence
- M_STOP to give up the I²C bus

If, during a read or write sequence, another M_READ or M_WRITE is requested by the ARM CPU, a restart is performed on the I²C bus. This opens the opportunity to provide a new slave device address in the MAster Address register before the command request.

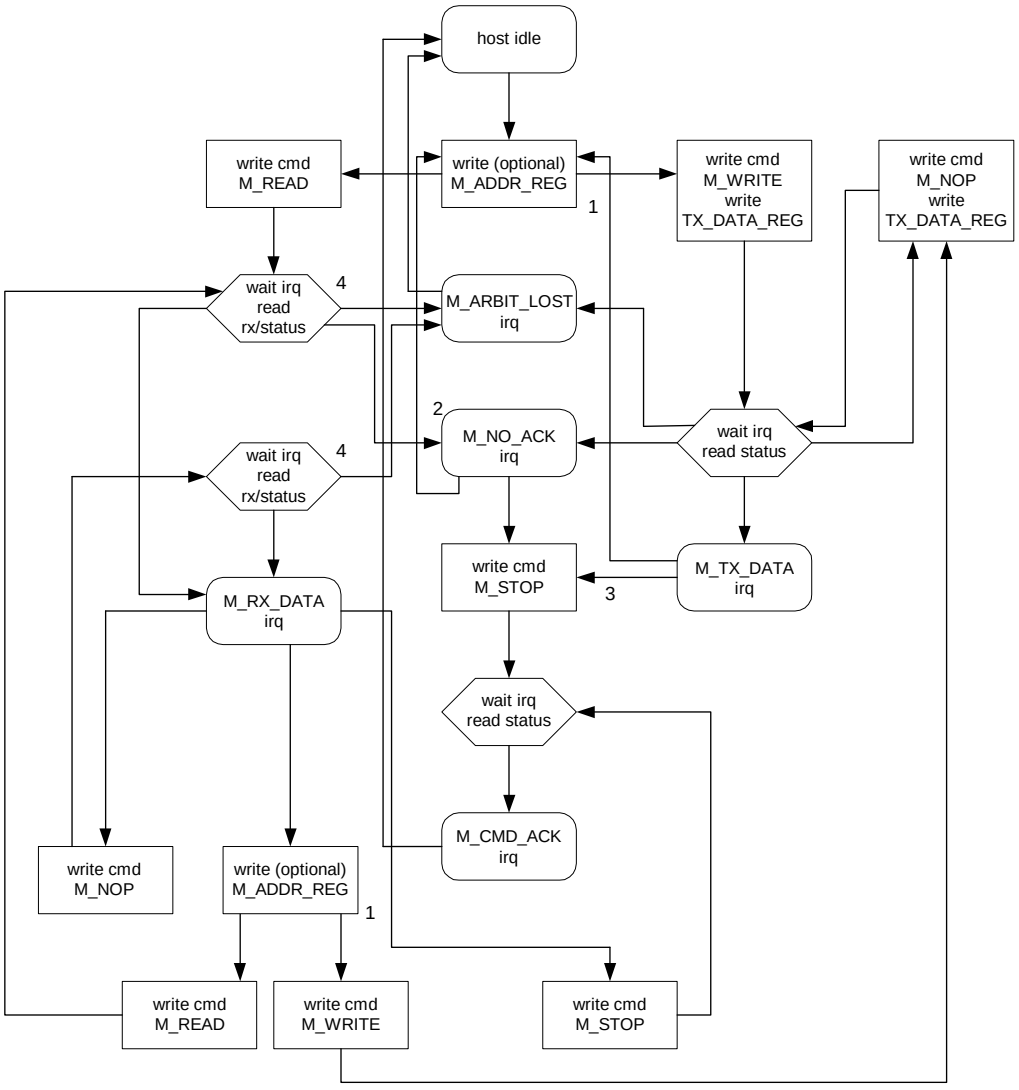
I²C slave high level driver

The I²C slave high level driver identifies one command: S_STOP, to discontinue a transaction. After this command, the slave remains inactive until the next start condition on the I²C bus. If a slave is accessed by a master, it generates S_RX_DATA and S_TX_DATA interrupts (see “Master/slave interrupt codes” on page 455). To distinguish the transactions from each other, special S_RX_DATA_1ST and S_TX_DATA_1ST interrupts are generated for the transmitted byte.

Downloaded from www.Manualslib.com manuals search engine

Flow charts

Master module (normal mode, 16-bit)

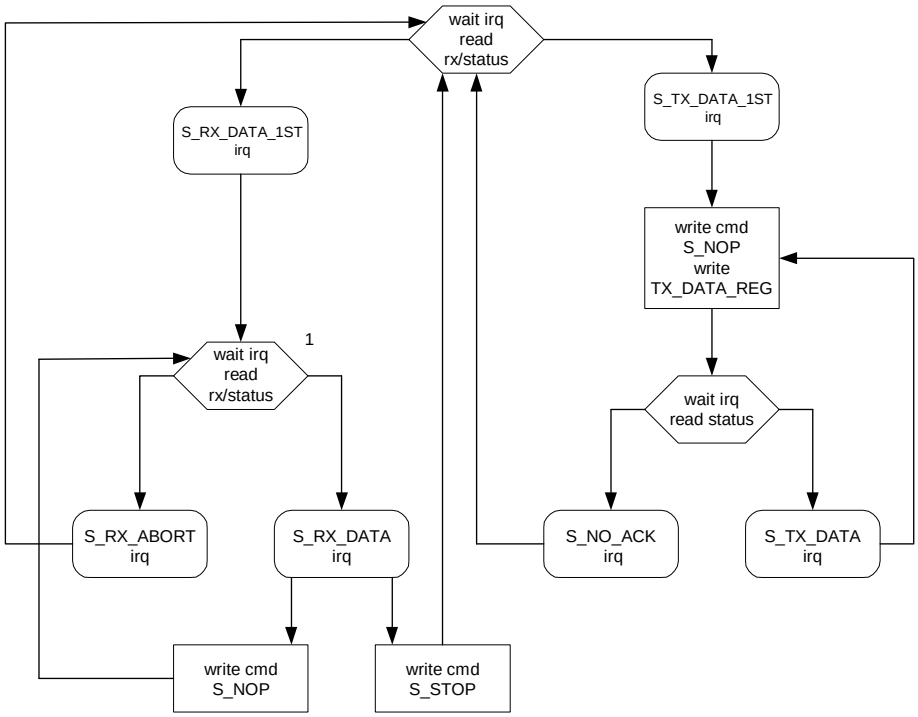


Notes:

1. Writing M_ADDR_REQ is not required if the device address is not changed.
2. Read on a non-existing slave.
3. Do not wait for the slave to perform a NO_ACK.
4. STATUS_REG and RX_DATA_REG are read simultaneously.

Downloaded from www.Manualslib.com manuals search engine

Slave module (normal mode, 16-bit)



Note: STATUS_REG and RX_DATA_REG are read simultaneously.

Downloaded from www.Manualslib.com manuals search engine

Real Time Clock Module

CHAPTER 14

The Real Time Clock (RTC) module tracks the time of the day to an accuracy of 10 milliseconds and provides calendar functionality that tracks day, month, and year.

RTC functionality

RTC monitors these time periods:

- Year from 1900-2999
- Month from 1-12
- Date from 1-28, 29, 30, or 31, as a function of year and month
- Day of week from 1-7
- Hour from 0-23, or from 1-12 with the AM/PM flag set
- Minute from 0-59
- Second from 0.00-59.99

RTC functionality also provides an alarm register that allows comparison of month, date, hour, minute, second, and hundredth-second. Each item can be masked, allowing an alarm to be generated at a particular time and date on a monthly basis. An interrupt can be generated on the alarm event.

Event detection finds and generates interrupts on rollover conditions, including rollovers into a new month, date, hour, minute, second, or hundredth-second.

Downloaded from www.Manualslib.com manuals search engine

RTC configuration and status registers

All configuration registers must be accessed as 32-bit words and as single accesses only. Bursting is not allowed.

Register address map

Address	Description
9006 0000	RTC General Control register
9006 0004	12/24 Hour register
9006 0008	Time register
90060 000C	Calendar register
9006 0010	Time Alarm register
9006 0014	Calendar Alarm register
9006 0018	Alarm Enable register
9006 001C	Event Flags register
9006 0020	Interrupt Enable register
9006 0024	Interrupt Disable register
9006 0028	Interrupt Status register
9006 002C	General Status register

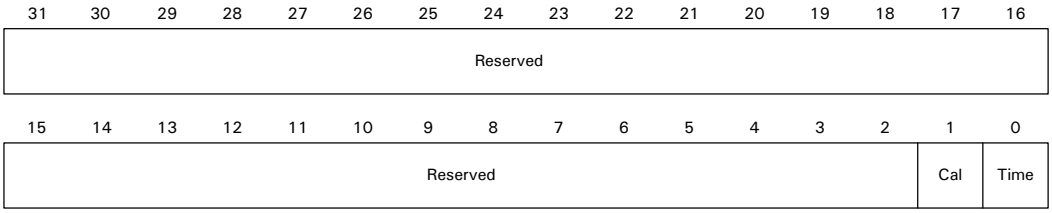
The reset values listed in the register descriptions are set when the regulated battery voltage on pins N3 and M4 drops below 1.56V.

RTC General Control register

Address: 9006 0000

The RTC General Control register contains miscellaneous settings for the RTC module.

Register



Downloaded from www.Manualslib.com manuals search engine

Register bit assignment

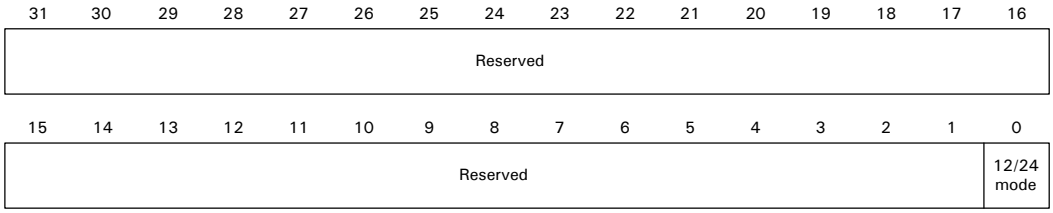
Bits	Access	Mnemonic	Reset	Description
D31:02	N/A	Reserved	N/A	N/A
D01	R/W	Cal	0x1	**Calendar operation** 0 Calendar operation enabled 1 Calendar operation disabled
D00	R/W	Time	0x1	**Time (date, hour, minute, second) operation** 0 Time operation enabled 1 Time operation disabled

12/24 Hour register

Address: 9006 0004

The 12/24 Hour register controls 12 or 24 hour clock mode operation.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:01	N/A	Reserved	N/A	N/A
D00	R/W	12/24	0x0	**12/24 clock mode operation** 0 24 hour mode operation 1 12 hour mode operation

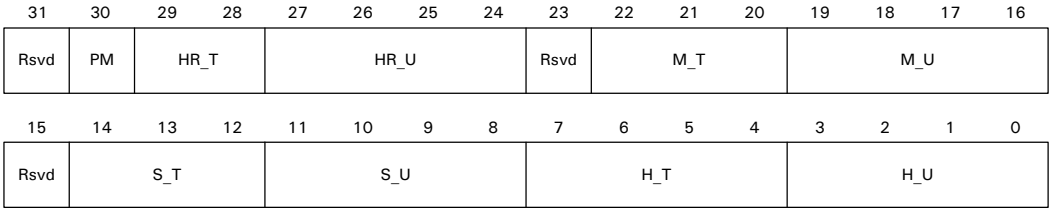
Downloaded from www.Manualslib.com manuals search engine

Time register

Address: 9006 0008

The TIme register sets the time values to the correct values, and reads the time registers. *BCD* is binary coded decimal.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31	N/A	Reserved	N/A	N/A
D30	R/W	PM	0x0	**PM** Used in 12 hour mode only. 0 AM 1 PM
D29:28	R/W	HR_T	0x0	Hours, tens, BCD digit (0-2)
D27:24	R/W	HR_U	0x0	Hours, units, BCD digit (0-9)
D23	N/A	Reserved	N/A	N/A
D22:20	R/W	M_T	0x0	Minutes, tens, BCD digit (0-5)
D19:16	R/W	M_U	0x0	Minutes, units, BCD digit (0-9)
D15	N/A	Reserved	N/A	N/A
D14:12	R/W	S_T	0x0	Seconds, tens, BCD digit (0-5)
D11:08	R/W	S_U	0x0	Seconds, units, BCD digit (0-9)
D07:04	R/W	H_T	0x0	Hundredths of a second, tens, BCD digit (0-9)
D03:00	R/W	H_U	0x0	Hundredths of a second, units, BCD digit (0-9)

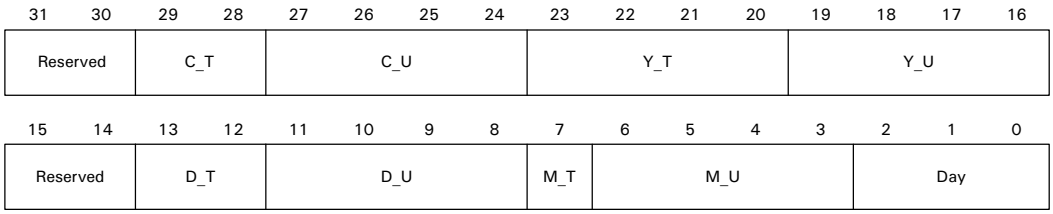
Downloaded from www.Manualslib.com manuals search engine

Calendar register

Address: 9006 000C

The Calendar register sets the calendar values to the correct values, and reads the calendar registers. *BCD* is binary coded decimal.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:30	N/A	Reserved	N/A	N/A
D29:28	R/W	C_T	0x0	Century, tens, BCD digit (1-2)
D27:24	R/W	C_U	0x0	Century, units, BCD digit (0-9)
D23:20	R/W	Y_T	0x0	Years, tens, BCD digit (0-9)
D19:16	R/W	Y_U	0x0	Years, units, BCD digit (0-9)
D15:14	N/A	Reserved	N/A	N/A
D13:12	R/W	D_T	0x0	Date, tens, BCD digit (0-3)
D11:08	R/W	D_U	0x0	Date, units, BCD digit (0-9)
D07	R/W	M_T	0x0	Months, tens, BCD digit (0-1)
D06:03	R/W	M_U	0x0	Months, units, BCD digit (0-9)
D02:00	R/W	Day	0x0	Day of week, units, BCD digit (0-7)

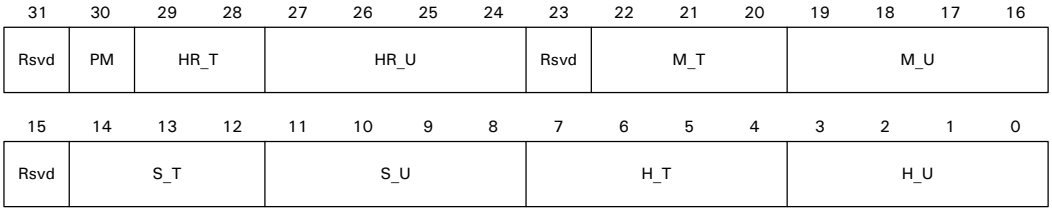
Downloaded from www.Manualslib.com manuals search engine

Time Alarm register

Address: 9006 0010

The Time Alarm register sets the time alarm. *BCD* is binary coded decimal.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31	N/A	Reserved	N/A	N/A
D30	R/W	PM	0x0	**PM** Used in 12 hour mode only. 0 AM 1 PM
D29:28	R/W	HR_T	0x0	Hours, tens, BCD digit (0-2)
D27:24	R/W	HR_U	0x0	Hours, units, BCD digit (0-9)
D23	N/A	Reserved	N/A	N/A
D22:20	R/W	M_T	0x0	Minutes, tens, BCD digit (0-5)
D19:16	R/W	M_U	0x0	Minutes, units, BCD digit (0-9)
D15	N/A	Reserved	N/A	N/A
D14:12	R/W	S_T	0x0	Seconds, tens, BCD digit (0-5)
D11:08	R/W	S_U	0x0	Seconds, units, BCD digit (0-9)
D07:04	R/W	H_T	0x0	Hundredths of a second, tens, BCD digit (0-9)
D03:00	R/W	H_U	0x0	Hundredths of a second, units, BCD digit (0-9)

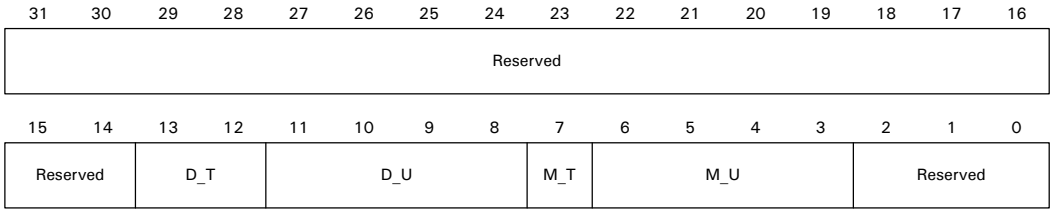
Downloaded from www.Manualslib.com manuals search engine

Calendar Alarm register

Address: 9006 0014

The Calendar Alarm register sets the calendar alarm. This register programs a specific date and month when an alarm should cause an event. You cannot set an alarm that is more than one year in the future. *BCD* is binary coded decimal.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:14	N/A	Reserved	N/A	N/A
D13:12	R/W	D_T	0x0	Date, tens, BCD digit (0-3)
D11:08	R/W	D_U	0x0	Date, units, BCD digit (0-9)
D07	R/W	M_T	0x0	Months, tens, CD digit (0-1)
D06:03	R/W	M_U	0x0	Months, units, BCD digit (0-9)
D02:00	N/A	Reserved	N/A	N/A

Alarm Enable register

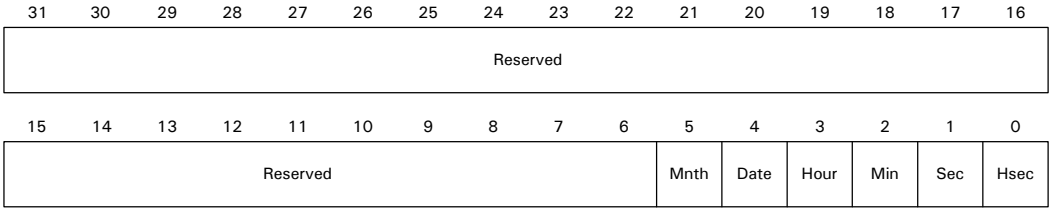
Address: 9006 0018

The Alarm Enable register sets the fields that can trigger an alarm. Setting a bit enables the corresponding time unit trigger event. Triggering the alarm causes an event to be generated, as set in the Events Flag register.

If all fields are enabled, an alarm is generated at the time set — the specific month, date, hour, minute, second, and hundredth-second. If only the minute field is set, the alarm triggers when that particular minute is reached, and every hour thereafter.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:06	N/A	Reserved	N/A	N/A
D05	R/W	Mnth	0x0	**Month** 0 Disable the month event 1 Enable the month event
D04	R/W	Date	0x0	**Date** 0 Disable the date event 1 Enable the date event
D03	R/W	Hour	0x0	**Hour** 0 Disable the hour event 1 Enable the hour event
D02	R/W	Min	0x0	**Minute** 0 Disable the minute event 1 Enable the minute event
D01	R/W	Sec	0x0	**Second** 0 Disable the second event 1 Enable the second event
D00	R/W	Hsec	0x0	**Hundredth of a second** 0 Disable the hundredth second event 1 Enable the hundredth second event

Event Flags register

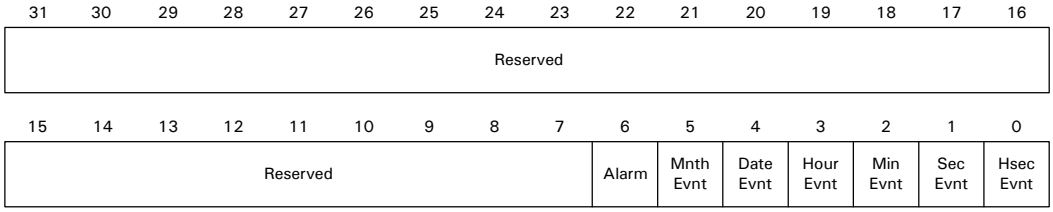
Address: 9006 001C

The Event Flags register indicates that an event has occurred since the last reset. Read the register to determine the cause of the current active interrupt. This register is cleared when read (*R/R* in Access column).

Note that the Event Flags register can change even if the corresponding alarm enable bit is not set.

Downloaded from www.Manualslib.com manuals search engine

Register


31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16
Reserved
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Reserved
Alarm
Mnth Evnt
Date Evnt
Hour Evnt
Min Evnt
Sec Evnt
Hsec Evnt

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:07	N/A	Reserved	N/A	N/A
D06	R/R	Alarm	0x0	**Alarm event** One of the events programmed in the Alarm Events register has triggered.
D05	R/R	Mnth Evnt	0x0	**Month event** 0 Month event has not occurred 1 Month event has occurred
D04	R/R	Date Evnt	0x0	**Date event** 0 Date event has not occurred 1 Date event has occurred
D03	R/R	Hour Evnt	0x0	**Hour event** 0 Hour event has not occurred 1 Hour event has occurred
D02	R/R	Min Evnt	0x0	**Minute event** 0 Minute event has not occurred 1 Minute event has occurred
D01	R/R	Sec Evnt	0x0	**Second event** 0 Second event has not occurred 1 Second event has occurred
D00	R/R	Hsec Evnt	0x0	**Hundredth of a second event** 0 Hundredth second event has not occurred 1 Hundredth second event has occurred

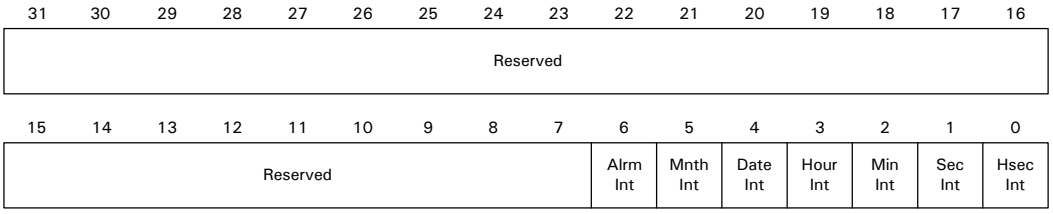
Downloaded from www.Manualslib.com manuals search engine

Interrupt Enable register

Address: 9006 0020

The Interrupt Enable register sets which events can generate and interrupt. The interrupt that is generated remains set until it is cleared by disabling the event or by reading/clearing the Event Flags register.

Register


31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16
Reserved
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Reserved
Alrm Int
Mnth Int
Date Int
Hour Int
Min Int
Sec Int
Hsec Int

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:07	N/A	Reserved	N/A	N/A
D06	W	Alrm Int	0x0	**Alarm interrupt** 0 Disable alarm interrupt 1 Enable alarm interrupt
D05	W	Mnth Int	0x0	**Month interrupt** 0 Disable month interrupt 1 Enable month interrupt
D04	W	Date Int	0x0	**Date interrupt** 0 Disable date interrupt 1 Enable date interrupt
D03	W	Hour Int	0x0	**Hour interrupt** 0 Disable hour interrupt 1 Enable hour interrupt
D02	W	Min Int	0x0	**Minute interrupt** 0 Disable minute interrupt 1 Enable minute interrupt
D01	W	Sec Int	0x0	**Second interrupt** 0 Disable second interrupt 1 Enable second interrupt
D00	W	Hsec Int	0x0	**Hundredth of a second interrupt** 0 Disable hundredth second interrupt 1 Enable hundredth second interrupt

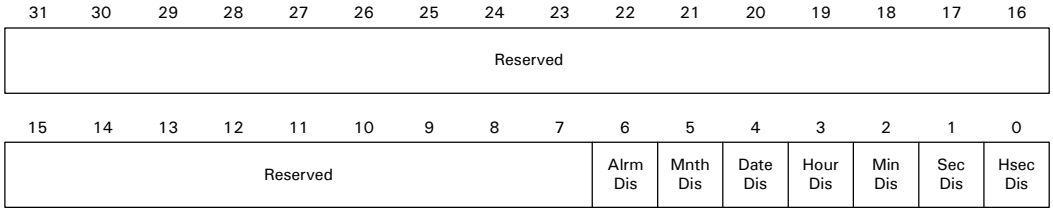
Downloaded from www.Manualslib.com manuals search engine

Interrupt Disable register

Address: 9006 0024

The Interrupt Disable register resets interrupts that are currently enables. An interrupt is disabled by writing a 1, then a 0, to the appropriate disable register bit.

Register


31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16
Reserved
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Reserved
Alrm Dis
Mnth Dis
Date Dis
Hour Dis
Min Dis
Sec Dis
Hsec Dis

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:07	N/A	Reserved	N/A	N/A
D06	W	Alrm Dis	0x0	**Alarm interrupt disable** 0 Enable alarm interrupt 1 Disable alarm interrupt
D05	W	Mnth Dis	0x0	**Month interrupt disable** 0 Enable month interrupt 1 Disable month interrupt
D04	W	Date Dis	0x0	**Date interrupt disable** 0 Enable date interrupt 1 Disable date interrupt
D03	W	Hour Dis	0x0	**Hour interrupt disable** 0 Enable hour interrupt 1 Disable hour interrupt
D02	W	Min Dis	0x0	**Minute interrupt disable** 0 Enable minute interrupt 1 Disable minute interrupt
D01	W	Sec Dis	0x0	**Second interrupt disable** 0 Enable second interrupt 1 Disable second interrupt
D00	W	Hsec Dis	0x0	**Hundredth of a second interrupt disable** 0 Enable hundredth second interrupt 1 Disable hundredth second interrupt

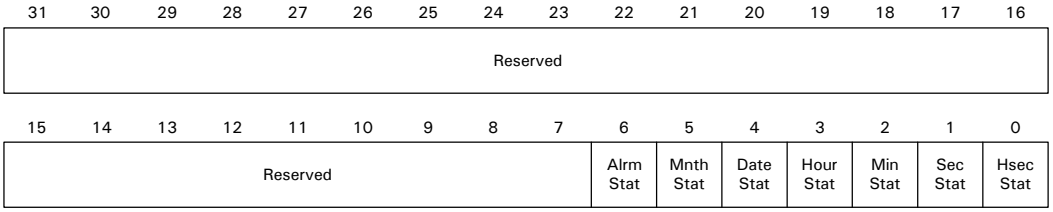
Downloaded from www.Manualslib.com manuals search engine

Interrupt Enable Status register

Address: 9006 0028

The Interrupt Enable Status register determines which interrupt sources are enabled and which interrupt sources are disabled.

Register


31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16
Reserved
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Reserved
Alrm Stat
Mnth Stat
Date Stat
Hour Stat
Min Stat
Sec Stat
Hsec Stat

Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:07	N/A	Reserved	N/A	N/A
D06	R	Alrm Stat	1	**Alarm interrupt status** 0 Interrupt enabled 1 Interrupt disabled
D05	R	Mnth Stat	1	**Month interrupt status** 0 Interrupt enabled 1 Interrupt disabled
D04	R	Date Stat	1	**Date interrupt status** 0 Interrupt enabled 1 Interrupt disabled
D03	R	Hour Stat	1	**Hour interrupt status** 0 Interrupt enabled 1 Interrupt disabled
D02	R	Min Stat	1	**Minute interrupt status** 0 Interrupt enabled 1 Interrupt disabled
D01	R	Sec Stat	1	**Second interrupt status** 0 Interrupt enabled 1 Interrupt disabled
D00	R	Hsec Stat	1	**Hundredth of a second interrupt status** 0 Interrupt enabled 1 Interrupt disabled

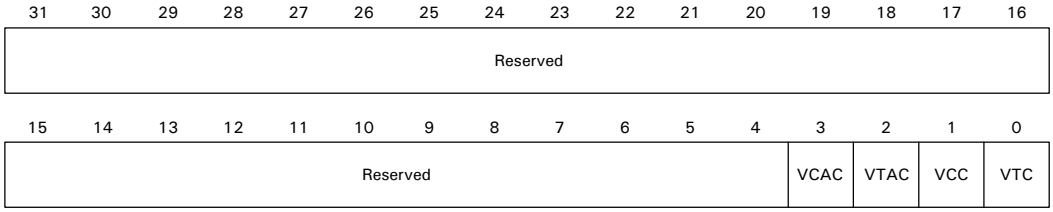
Downloaded from www.Manualslib.com manuals search engine

General Status register

Address: 9006 002C

The General Status register determines the status of the RTC configuration. If an invalid configuration is found, the RTC counters do not start operation.

Register



Register bit assignment

Bits	Access	Mnemonic	Reset	Description
D31:04	N/A	Reserved	N/A	N/A
D03	R	VCAC	0x1	**Valid calendar alarm configuration** 0 Invalid 1 Valid
D02	R	VTAC	0x1	**Valid time alarm configuration** 0 Invalid 1 Valid
D01	R	VCC		**Valid calendar configuration** 0 Invalid 1 Valid
D00	R	VTC		**Valid time configuration** 0 Invalid 1 Valid

Downloaded from www.Manualslib.com manuals search engine

Downloaded from www.Manualslib.com manuals search engine

Analog-to-Digital Converter (ADC) Module

CHAPTER 15

The NS9215 ASIC supports a 12-bit successive approximation analog-to-digital converter (ADC). To maximize flexibility, an input pin is provided to apply an external reference voltage, which defines the full scale input range. An analog multiplexer is included to enable the selection of up to eight inputs.

Features

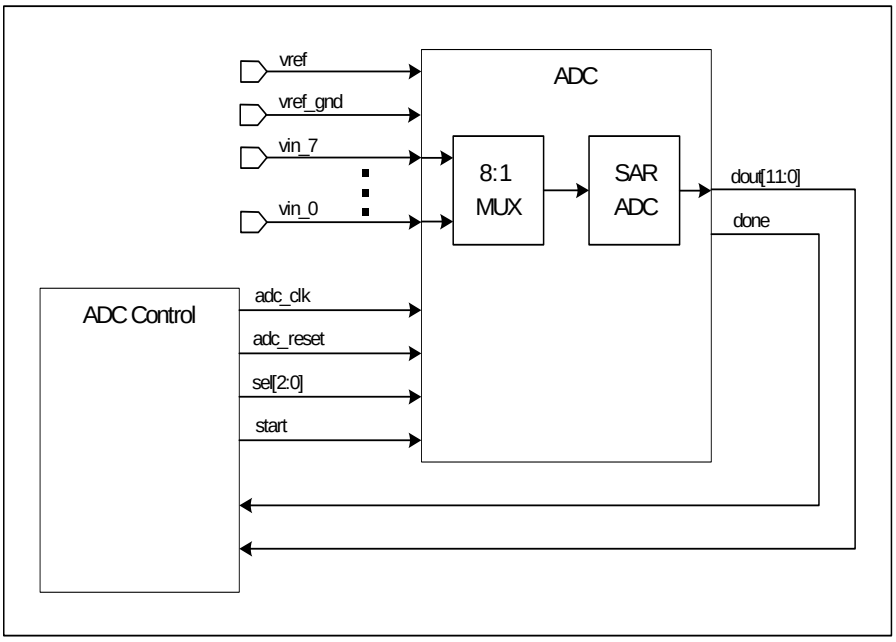
The ADC module supports these features:

- 12-bit resolution
- 1 MHz conversion rate
- Single-ended 8:1 multiplexed inputs
- Rail-to-rail input range
- 12-bit output, either DMA or direct CPU access

ADC module structure

This diagram shows the ADC module structure.

Downloaded from www.Manualslib.com manuals search engine



ADC control block

The ADC control block provides access between the CPU and the ADC module. The ADC clock and control signals are generated in this block. The ADC module output can be either DMA'd to memory or read directly by the CPU.

- If DMA is enabled, ADC output data is written to memory using UART D's receive DMA controller.
- If more than one channel is enabled, word 0 in the DMA buffer will always be from channel 0, followed by the data from the other selected channels.
- The data buffer length must be a word multiple of the number of selected channels. For example, if three channels are selected, the buffer length must be a multiple of three words or 12 bytes.

ADC DMA procedure

If using DMA, the DMA channel must be set up first and enabled before enabling the ADC. The procedure below must be followed each time a new DMA is started or if a DMA FIFO overflow is detected. The RX FIFO overflow interrupt should be enabled to detect an overflow.

1 Configure the ADC Configuration register at address 9003 9000 for DMA operation (bit 3 set to 1) and the number of channels but leave bit 31 set to a 0.

Downloaded from www.Manualslib.com manuals search engine

2 Set up the ADC DMA control registers and buffer descriptors (UART channel D).

3 Reset the ADC module by writing a 0 then a 1 to bit 8 in the Module Reset register at address A090 0180.

4 Flush the ADC DMA FIFO by writing a 1 then a 0 to bit 17 in UART Channel D Wrapper Configuration register at address 9002 9000.

5 Enable the ADC DMA channel by writing a 1 then a 0 to bit 31 in the UART D DMA RX Control register at address 9002 8004. 6. Start the ADC by writing a 1 to bit 31 in the ADC Configuration register at address 9003 9000.

ADC control and status registers

The ADC configuration registers are located at offset 0x9003_9000.

Register address map

Address	Description	Access	Reset value
9003_9000	ADC Configuration register	R/W	0x00000000
9003_9004	ADC Clock Configuration register	R/W	0x00000000
9003_9008	ADC Output 0 register	R/W	0x00000000
9003_900C	ADC Output 1 register	R/W	0x00000000
9003_9010	ADC Output 2 register	R/W	0x00000000
9003_9014	ADC Output 3 register	R/W	0x00000000
9003_9018	ADC Output 4 register	R/W	0x00000000
9003_901C	ADC Output 5 register	R/W	0x00000000
9003_9020	ADC Output 6 register	R/W	0x00000000
9003_9024	ADC Output 7 register	R/W	0x00000000

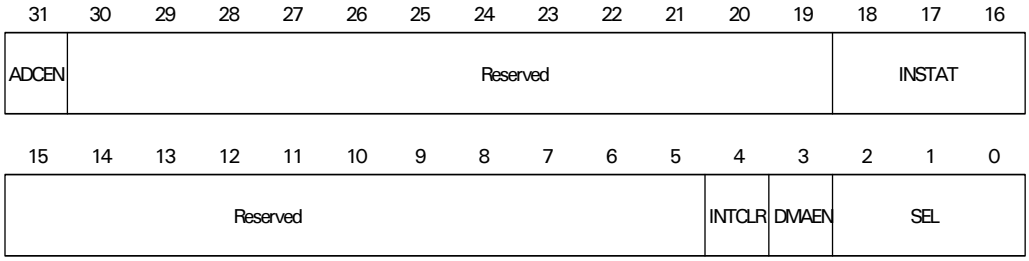
ADC Configuration register

Address: 9003_9000

The ADC Configuration register is the primary Wrapper Configuration register.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

Bit(s)	Access	Mnemonic	Reset	Description
D31	R/W	ADCEN	0	0 The ADC module is disabled and held in reset 1 The ADC module is enabled
D30:19	N/A	Reserved	N/A	N/A
D18:16	R	INSTAT	0	**Interrupt status** Indicates the channel processed at the time of the interrupt.
D15:5	N/A	Reserved	N/A	N/A
D04	R/W	INTCLR	0	**Interrupt clear** The ADC module generates an interrupt each time the ADC generates a new value. This bit clears the interrupt. The CPU must write a 1, then a 0 to this bit to clear the interrupt.
D03	R/W	DMAEN	0	**DMA enable** If set, ADC output data is written to memory using UART D's receive DMA. 0 DMA disabled 1 DMA enabled
D02:00	R/W	SEL	000	**ADC channel select** Controls how many channels are active. 000 Channel 0 001 Channels 0-1 010 Channels 0-2 011 Channels 0-3 100 Channels 0-4 101 Channels 0-5 110 Channels 0-6 111 Channels 0-7

Downloaded from www.Manualslib.com manuals search engine

ADC Clock Configuration register

Address: 9003_9004

The ADC Clock Configuration register controls the ADC clock generator. The source clock is the output of the PLL. The maximum ADC clock frequency is 14 MHz and the conversion time is 14 clock cycles. This is the formula for the ADC clock:

ADC clock = PLL clock / (2 x (N+1))

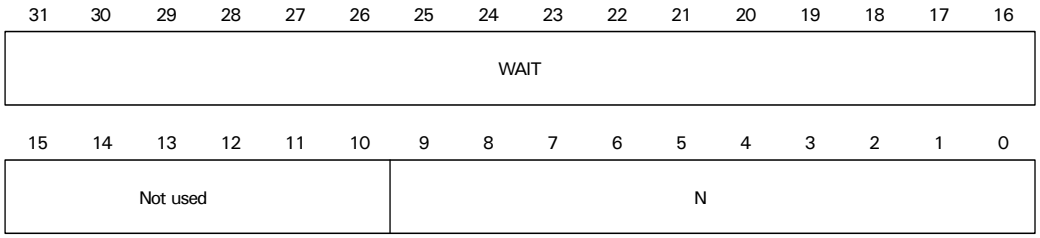
Example

- PLL clock frequency = 299.8272 MHz
- N value = 10
- ADC clock frequency:

  ADC clock = 299.8272 MHz / (2 x (10+1)) = 13.6285 MHz

Wait states can be added to increase conversion time beyond 14 clock cycles.

Register



Register bit assignment

Bit(s)	Access	Mnemonic	Reset	Description
D31:16	R/W	WAIT	N/A	Number of additional clock cycles per conversion cycle.
D15:10	R/W	Not used	0	This field must be written to 0.
D09:00	R/W	N	0	ADC clock converter.

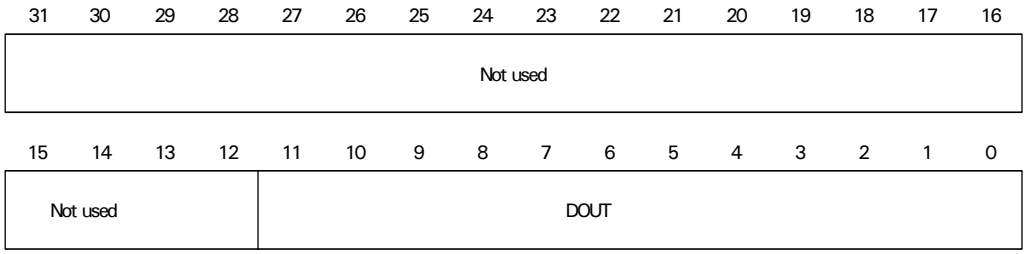
ADC Output Registers 0-7

Addresses: 9003_9008 / 9003_900C / 9003_9010 / 9003_9014 / 9003_9018 / 9003_901C / 9003_9020 / 9003_9024

The ADC Output registers provide CPU access for the ADC output for each channel.

Downloaded from www.Manualslib.com manuals search engine

Register



Register bit assignment

Bit(s)	Access	Mnemonic	Reset	Description
D31:12	R/W	Not used	0	This field must be written to 0.
D11:00	R	DOUT	0	Provides the output of the ADC for each channel.

Downloaded from www.Manualslib.com manuals search engine

Timing

CHAPTER 16

This chapter provides the electrical specifications, or timing, integral to the operation of the processor. Timing includes information about DC and AC characteristics, output rise and fall timing, and crystal oscillator specifications.

Electrical characteristics

Absolute maximum ratings

The processor operates at a 1.8V core, with 3.3V I/O ring voltages.

Permanent device damage can occur if absolute maximum ratings are ever exceeded. Absolute maximum ratings are below.

Parameter	Symbol[a]	Rating	Unit
DC supply voltage	V_{DDA}	- 0.3 to + 3.9	V
DC input voltage	V_{INA}	-0.3 to 5.0V	V
DC output voltage	V_{OUTA}	-0.3 to V_{DDA}+0.3	V
DC input current	I_{IN}	± 10	mA
Storage temperature	T_{STG}	- 40 to + 125	°C

aV_{DDA}, V_{INA}, V_{OUTA}: Ratings of I/O cells for 3.3V interface.
V_{DDC}: Ratings of internal cell.
The processor is immune to power supply sequencing problems.

Downloaded from www.Manualslib.com manuals search engine

Recommended operating conditions

Recommended operating conditions specify voltage and temperature ranges over which a circuit's correct logic function is guaranteed. The specified DC electrical characteristics are satisfied over these ranges. Below are the recommended operating conditions.

Parameter	Symbol[a]	Rating	Unit
DC supply voltage	V_{DDA}	3.0 to 3.6	V
	V_{DDC} (core)	1.62 to 1.98	V
Maximum junction temperature	T_J	125	^oC

a V_{DDA}: Ratings of I/O cells for 3.3V interface.
$V_{DDC:}$ Ratings of internal cells

Power dissipation

The table below shows the maximum power dissipation for I/O and core:

CPU / Memory clock	Power
150MHz/75MHz	Total 1.019W Core 0.880W I/O 0.139W
75 MHz/75MHz	Total 0.828W Core 0.696W I/O 0.132W
112MHz/56MHz	Total 0.638W Core 0.536W I/O 0.102W
56MHz/56MHz	Total 0.499W Core 0.403W I/O 0.096W
Sleep Mode, wake on Ethernet	Total 0.073W Core 0.027W I/O 0.046W
Sleep Mode, wake on External IRQ	Total 0.055W Core 0.022W I/O 0.033W
Main Power Down, Battery Draw	3.0V - 32uA 1.8V - 6uA

Downloaded from www.Manualslib.com manuals search engine

DC electrical characteristics

DC characteristics specify the worst-case DC electrical performance of the I/O buffers that are guaranteed over the specified temperature range.

Inputs

All electrical inputs are 3.3V interface.

The processor I/O are 5 volt tolerant.

DC electrical inputs are provided below.

Sym	Parameter	Condition[a]	Value		Unit
V_{IH}	High-level input voltage: LVTTL level		Min	2.0	V
V_{IL}	Low-level input voltage: LVTTL level		Max	0.8	V
I_{IH}	High level input current (no pulldown) Input buffer with pulldown	$V_{INA} = V_{DDA}$	Min/Max	-10/10	µA
			Min/Max	10/200	µA
I_{IL}	Low-level input current (no pullup Input buffer with pullup	$V_{INA} = V_{SS}$	Min/Max	-10/10	µA
			Min/Max	10/200	µA
I_{OZ}	High-impedance leakage current	$V_{OUTA} = V_{DDA}$ or V_{SS}	Min/Max	-10/10	µA

a V_{SS} = 0V (GND)

Downloaded from www.Manualslib.com manuals search engine

Ouputs

All electrical outputs are 3.3V interface.

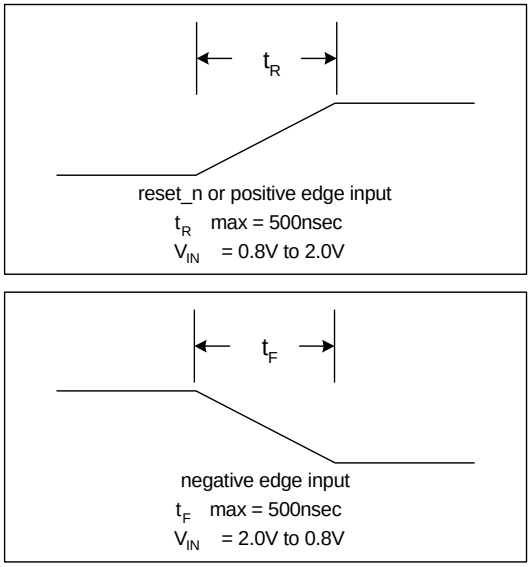
DC electrical outputs are provided below.

Sym	Parameter	Value		Unit
V_{OH}	High-level output voltage (LVTTL level)	Min	V_{DDA}-0.6	V
V_{IL}	Low-level input voltage: LVTTL level	Max	0.4	V

Reset and edge sensitive input timing requirements

The critical timing requirement is the rise and fall time of the input. If the rise time is too slow for the reset input, the hardware strapping options may be registered incorrectly. If the rise time of a positive-edge-triggered external interrupt is too slow, then an interrupt may be detected on both the rising and falling edge of the input signal.

A maximum rise and fall time must be met to ensure that reset and edge sensitive inputs are handled correctly. With Digi processors, the maximum is 500 nanoseconds as shown:



Downloaded from www.Manualslib.com manuals search engine

If an external device driving the reset or edge sensitive input on a Digi processor cannot meet the 500ns maximum rise and fall time requirement, the signal must be buffered with a Schmitt trigger device. Here are sample Schmitt trigger device part numbers:

Manufacturer	Part Number	Description
Fairchild	NC7SP17	Single Schmitt trigger buffer, available in 5-lead SC70 and 6-lead MicroPak packages
Philips	74LVC1G17GW	Single Schmitt trigger buffer, available in 5-lead SC70 and SOT 353 packages
TI	SN74LVC1G17DC	Single Schmitt trigger buffer, available in 5-lead SC70 and SOT 353 packages
ON Semi	NL17SZ17DFT2	Single Schmitt trigger buffer, available in 5-lead SC70 and SOT 353 packages.

Downloaded from www.Manualslib.com manuals search engine

Memory Timing

All AC characteristics are measured with 35pF, unless otherwise noted.

Memory timing contains parameters and diagrams for both SDRAM and SRAM timing.

The table below describes the values shown in the SDRAM timing diagrams.

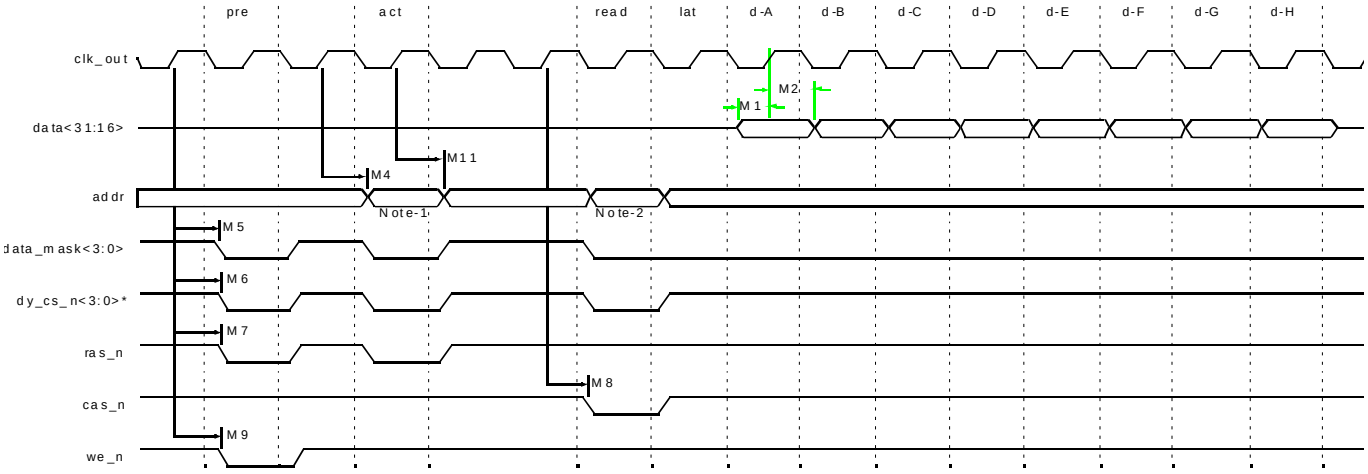
Parm	Description	Min	Max	Unit	Notes
M1	data input setup time to rising	1.0		ns	
M2	data input hold time to rising	0.0		ns	
M4	clk_out high to address valid		9.5	ns	
M11	address hold time	4.0			
M5	clk_out high to data_mask		9.5	ns	1, 2
M6	clk_out high to dy_cs_n low		9.5	ns	3, 4
M7	clk_out high to ras_n low		9.5	ns	
M8	clk_out high to cas_n low		9.5	ns	
M9	clk_out high to we_n low		9.5	ns	
M10	clk_out high to data out		9.5	ns	
M12	data out hold time	4.0			
M3	clk_out high to clk_en high		9.5	ns	
M13	clk_en high to sdram access	2	2	clock	
M14	end sdram access to clk_en low	2	2	clocks	

Notes:

1 All four data_mask signals are used for all transfers.
2 All four data_mask signals will go low during a read cycle, for both 16-bit and 32-bit transfers.
3 Only one of the clk_out signals is used.
4 Only one of the dy_cs_n signals is used.

Downloaded from www.Manualslib.com manuals search engine

SDRAM burst read (16-bit)



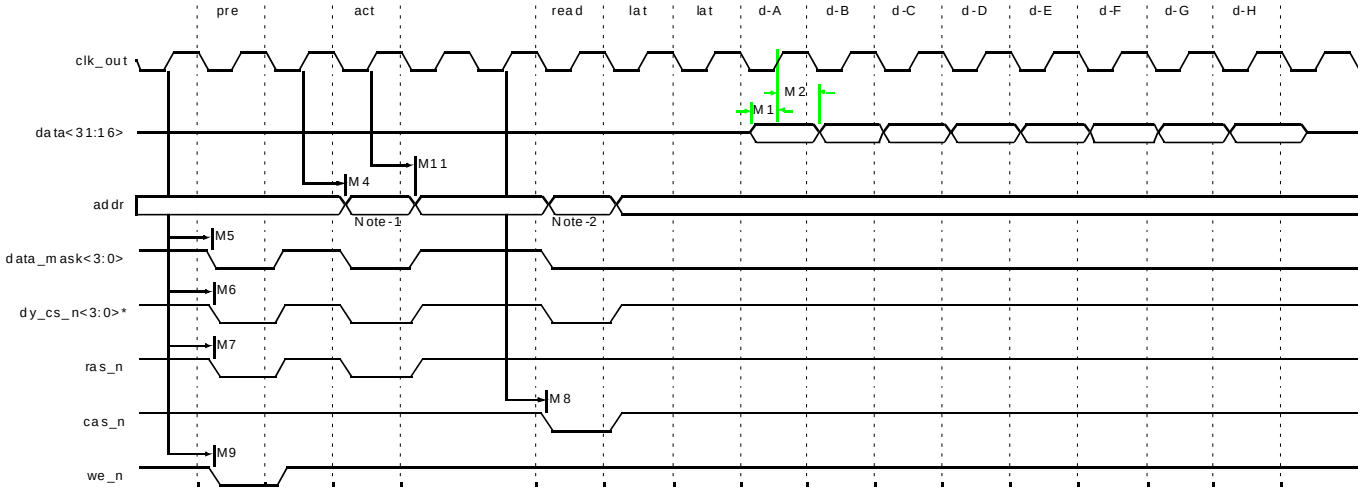
Notes:

1 This is the bank and RAS address.

2 This is the CAS address.

Downloaded from www.Manualslib.com manuals search engine

SDRAM burst read (16 bit), CAS latency = 3



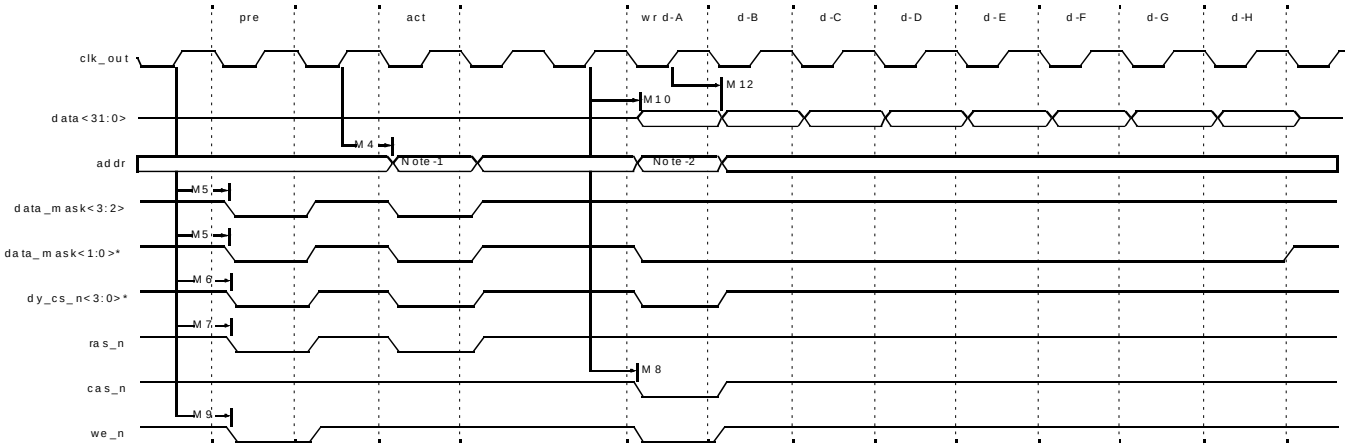
Notes:

1 This is the bank and RAS address.

2 This is the CAS address.

Downloaded from www.Manualslib.com manuals search engine

SDRAM burst write (16 bit)



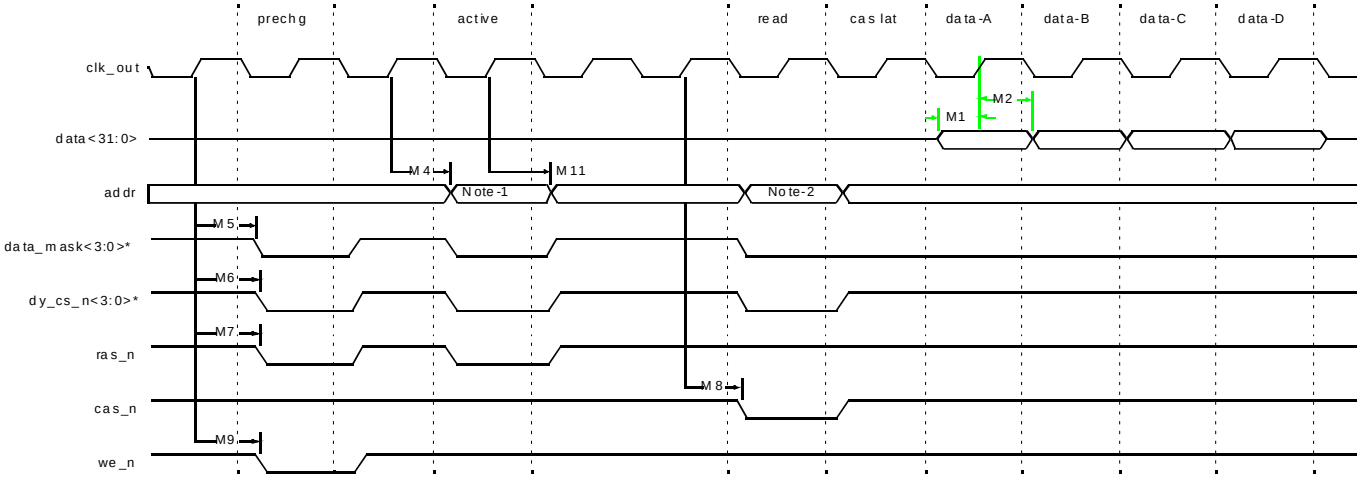
Notes:

1 This is the bank and RAS address.

2 This is the CAS address.

Downloaded from www.Manualslib.com manuals search engine

SDRAM burst read (32 bit)



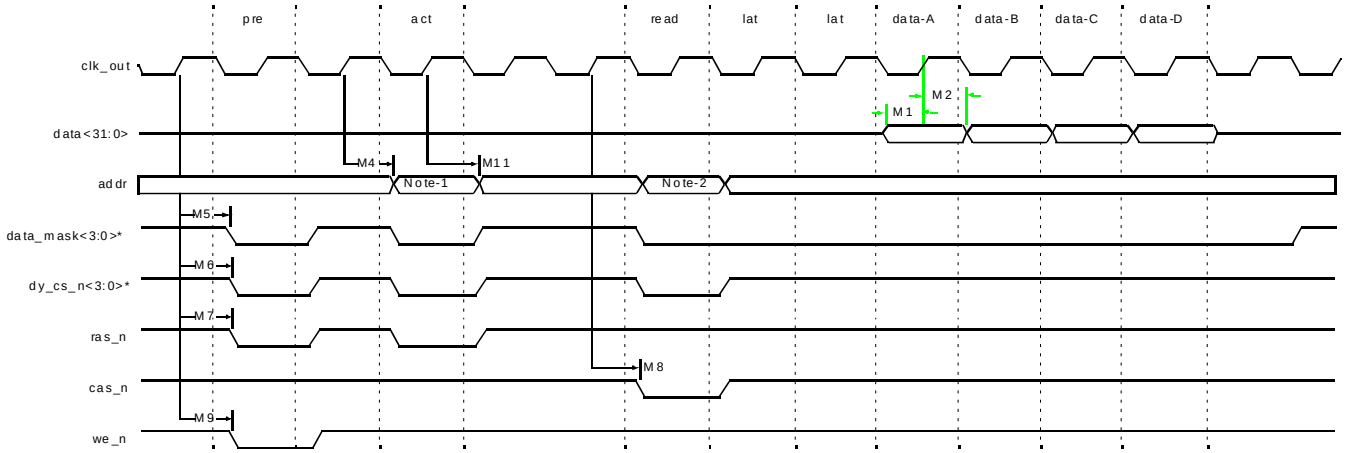
Notes:

1 This is the bank and RAS address.

2 This is the CAS address.

Downloaded from www.Manualslib.com manuals search engine

SDRAM burst read (32 bit), CAS latency = 3



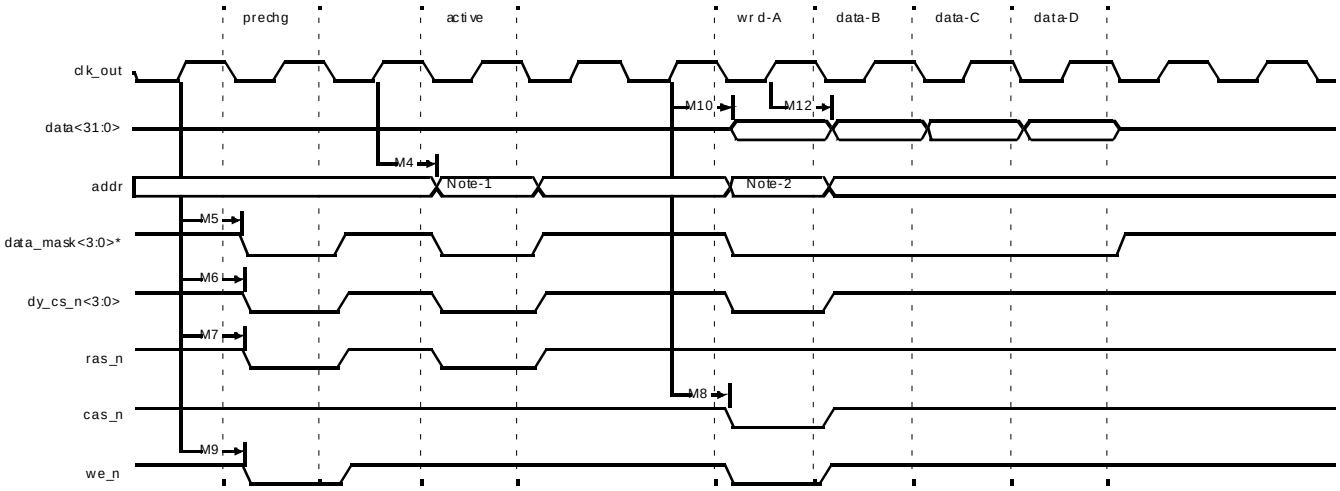
Notes:

1 This is the bank and RAS address.

2 This is the CAS address.

Downloaded from www.Manualslib.com manuals search engine

SDRAM burst write (32-bit)



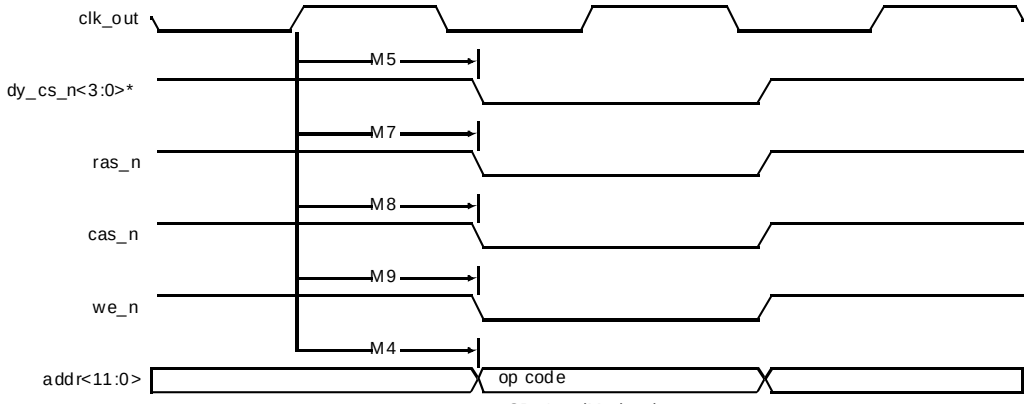
Notes:

1 This is the bank and RAS address.

2 This is the CAS address.

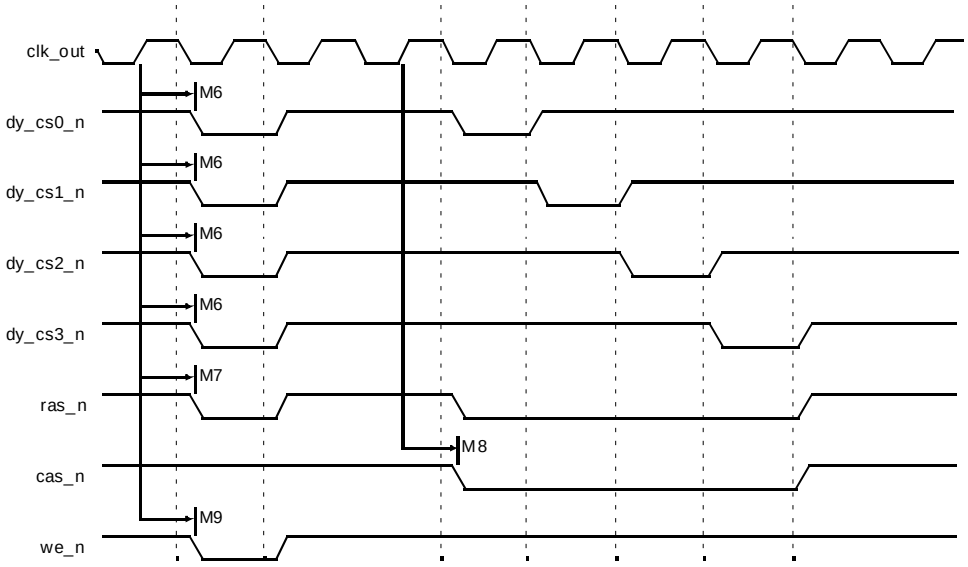
Downloaded from www.Manualslib.com manuals search engine

SDRAM load mode



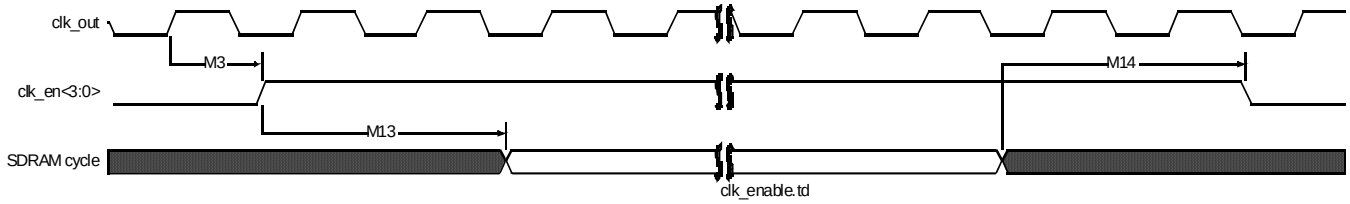
Downloaded from www.Manualslib.com manuals search engine

SDRAM refresh mode



Downloaded from www.Manualslib.com manuals search engine

Clock enable timing



Downloaded from www.Manualslib.com manuals search engine

Values in SRAM timing diagrams

The next table describes the values shown in the SRAM timing diagrams.

Parm	Description	Min	Max	Unit	Notes
M15	clock high to data out valid	-2	+2	ns	
M16	data out hold time from clock high	-2	+2	ns	
M17	clock high to address valid	-2	+2	ns	
M18	address hold time from clock high	-2	+2	ns	
M19	clock high to st_cs_n low	-2	+2	ns	2
M20	clock high to st_cs_n high	-2	+2	ns	2
M21	clock high to we_n low	-2	+2	ns	
M22	clock high to we_n high	-2	+2	ns	
M23	clock high to byte_lanes low	-2	+2	ns	
M24	clock high to byte_lanes high	-2	+2	ns	
M25	data input setup time to rising clk	10		ns	
M26	data input hold time to rising clk	0		ns	
M27	clock high to oe_n low	-2	+2	ns	
M28	clock high to oe_n high	-2	+2	ns	

Notes:

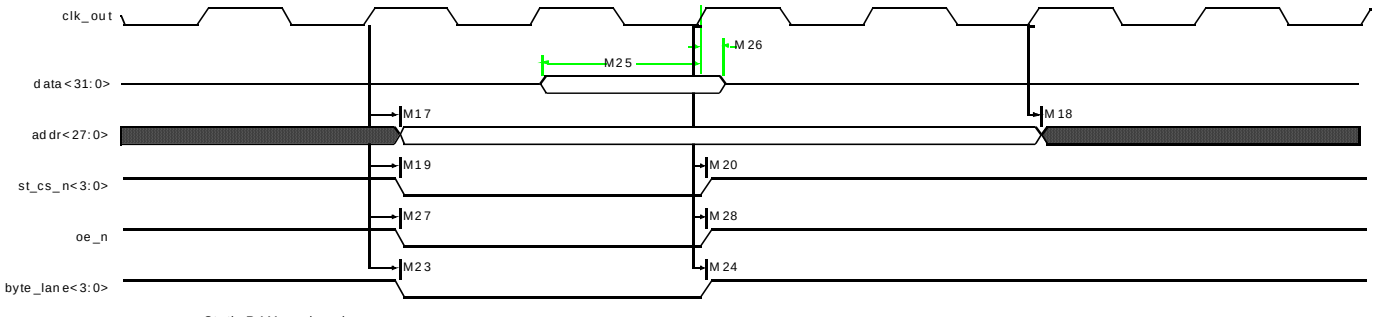
1 The (CPU clock out / 2) signal is for reference only.

2 Only one of the four dy_cs_n signals is used. The diagrams show the active low configuration, which can be reversed (active high) with the PC field.

3 Use this formula to calculate the length of the st_cs_n signal: Tacc + board delay + (optional buffer delays, both address out and data in) + 10ns

Downloaded from www.Manualslib.com manuals search engine

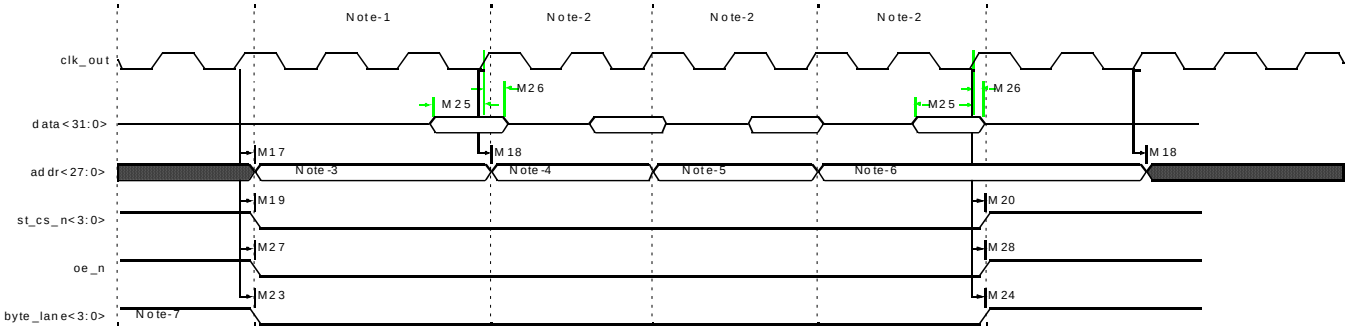
Static RAM read cycles with 0 wait states



- WTRD = 1
- WOEN = 0
- If the PB field is set to 1, all four byte_lane signals will go low for 32-bit, 16-bit, and 8-bit read cycles.
- If the PB field is set to 0, the byte_lane signal will always be high.

Downloaded from www.Manualslib.com manuals search engine

Static RAM asynchronous page mode read, WTPG = 1



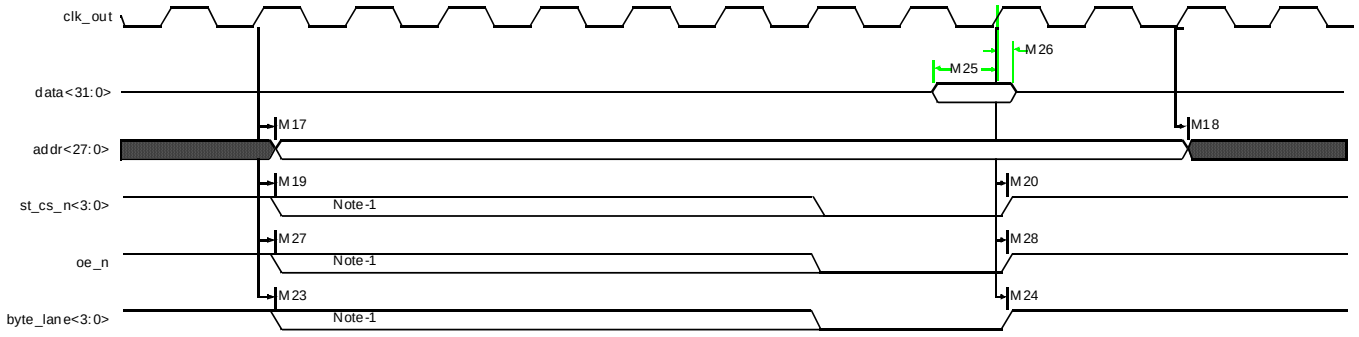
- WTPG = 1
- WTRD = 2
- If the PB field is set to 1, all four byte_lane signals will go low for 32-bit,
- 16-bit, and 8-bit read cycles.
- The asynchronous page mode will read 16 bytes in a page cycle. A 32-bit bus will do four 32-bit reads, as shown (3-2-2-2). A 16-bit bus will do eight 16-bit reads (3-2-2-2-3-2-2-2) per page cycle, and an 8-bit bus will do sixteen 8-bit reads (3-2-2-2-3-2-2-2-3-2-2-2-3-2-2-2) per page cycle. 3-2-2-2 is the example used here, but the WTRD and WTPG fields can set them differently.

Notes:

1 The length of the first cycle in the page is determined by the WTRD field.
2 The length of the 2nd, 3rd, and 4th cycles is determined by the WTPG field.
3 This is the starting address. The least significant two bits will always be '00.'
4 The least significant two bits in the second cycle will always be '01.'
5 The least significant two bits in the third cycle will always be '10.'
6 The least significant two bits in the fourth cycle will always be '11.'
7 If the PB field is set to 0, the byte_lane signal will always be high during a read cycle.
8 Setting the BMODE (Burst mode) bit D02 in the static memory configuration register allows the static output enable signal to toggle during bursts.

Downloaded from www.Manualslib.com manuals search engine

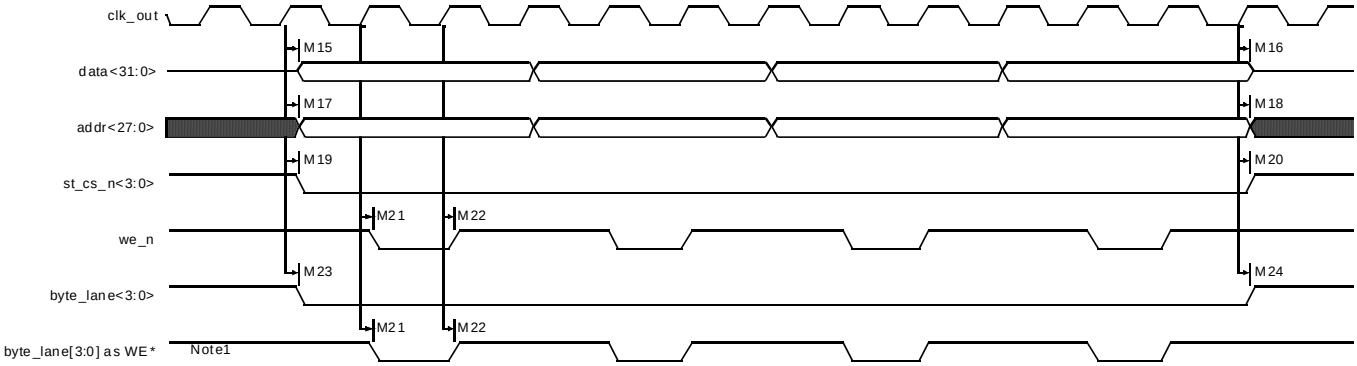
Static RAM read cycle with configurable wait states



- WTRD = from 1 to 15
- WOEN = from 0 to 15
- If the PB field is set to 1, all four byte_lane signals will go low for 32-bit, 16-bit, and 8-bit read cycles.
- If the PB field is set to 0, the byte_lane signal will always be high.

Downloaded from www.Manualslib.com manuals search engine

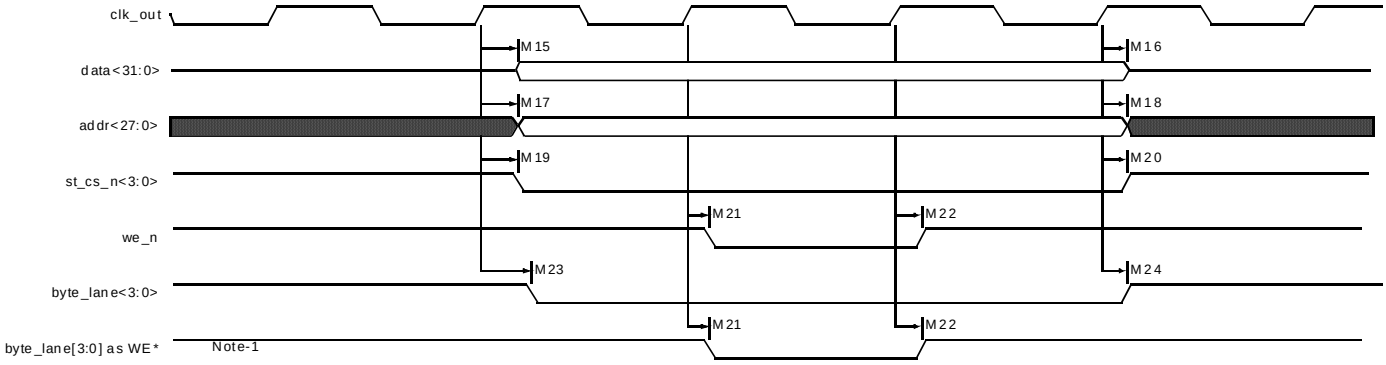
Static RAM sequential write cycles



- WTWR = 0
- WWEN = 0
- During a 32-bit transfer, all four byte_lane signals will go low.
- During a 16-bit transfer, two byte_lane signals will go low.
- During an 8-bit transfer, only one byte_lane signal will go low.
- Note:
- If the PB field is set to 0, the byte_lane signals will function as write enable signals and the we_n signal will always be high.

Downloaded from www.Manualslib.com manuals search engine

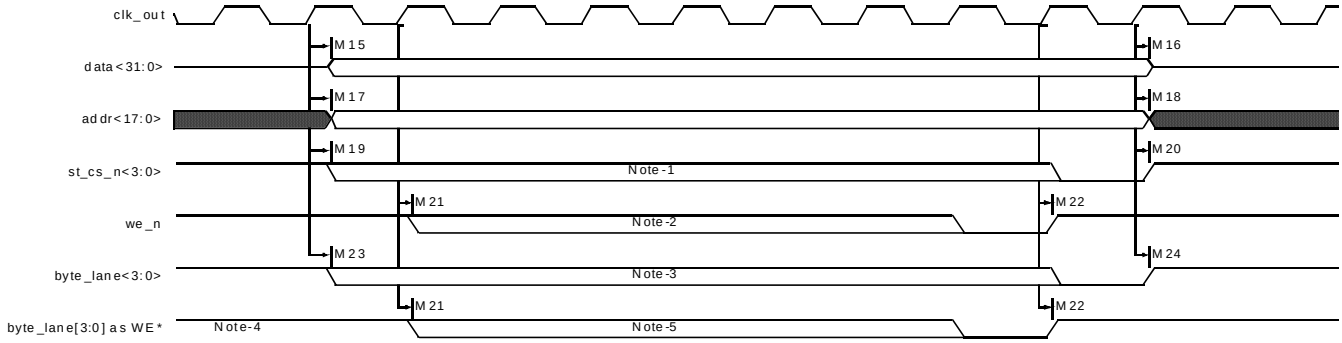
Static RAM write cycle



- WTWR = 0
- WWEN = 0
- During a 32-bit transfer, all four byte_lane signals will go low.
- During a 16-bit transfer, two byte_lane signals will go low.
- During an 8-bit transfer, only one byte_lane signal will go low.
- Note:
- If the PB field is set to 0, the byte_lane signals will function as write enable signals and the we_n signal will always be high.

Downloaded from www.Manualslib.com manuals search engine

Static write cycle with configurable wait states



- WTWR = from 0 to 15
- WWEN = from 0 to 15
- The WTWR field determines the length on the write cycle.
- During a 32-bit transfer, all four byte_lane signals will go low.
- During a 16-bit transfer, two byte_lane signals will go low.
- During an 8-bit transfer, only one byte_lane signal will go low.

Notes:

1 Timing of the st_cs_n signal is determined with a combination of the WTWR and WWEN fields. The st_cs_n signal will always go low at least one clock before we_n goes low, and will go high one clock after we_n goes high.

2 Timing of the we_n signal is determined with a combination of the WTWR and WWEN fields.

3 Timing of the byte_lane signals is determined with a combination of the WTWR and WWEN fields. The byte_lane signals will always go low one clock before we_n goes low, and will go one clock high after we_n goes high.

4 If the PB field is set to 0, the byte_lane signals will function as the write enable signals and the we_n signal will always be high.

5 If the PB field is set to 0, the timing for the byte_lane signals is set with the WTWR and WWEN fields.

Downloaded from www.Manualslib.com manuals search engine

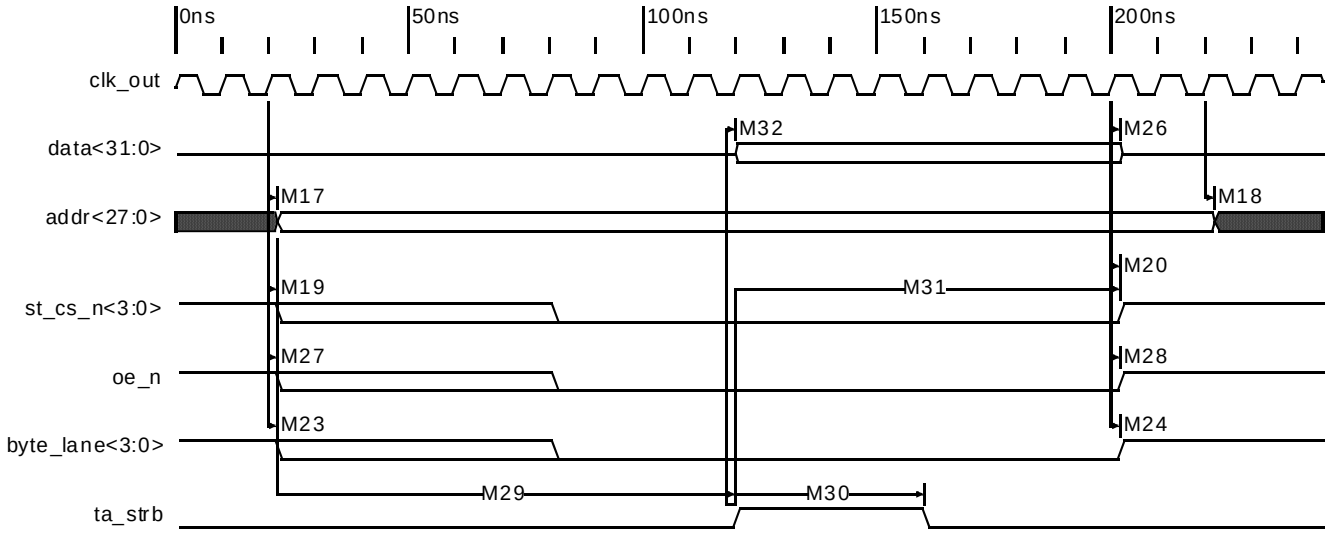
Slow peripheral acknowledge timing

The table below describes the values shown in the slow peripheral acknowledge timing diagrams.

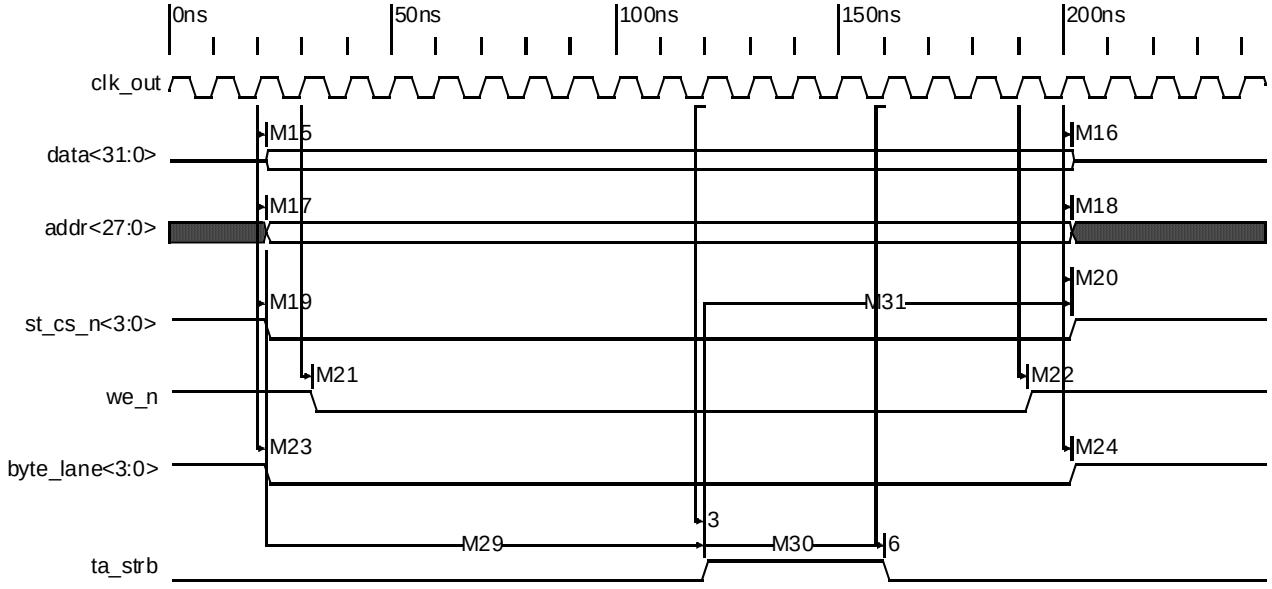
Parm	Description	Min	Max	Unit	Notes
M15	clock high to data out valid	-2	+2	ns	
M16	data out hold time from clock high	-2	+2	ns	
M17	clock high to address valid	-2	+2	ns	
M18	address hold time from clock high	-2	+2	ns	
M19	clock high to st_cs_n low	-2	+2	ns	2
M20	clock high to st_cs_n high	-2	+2	ns	2
M21	clock high to we_n low	-2	+2	ns	
M22	clock high to we_n high	-2	+2	ns	
M23	clock high to byte_lanes low	-2	+2	ns	
M24	lock high to byte_lanes high	-2	+2	ns	
M26	data input hold time to rising clk	0		ns	
M27	clock high to oe_n low	-2	+2	ns	
M28	clock high to oe_n high	-2	+2	ns	
M29	address/chip select valid to ta_strb high	2		CPU cycles	
M30	ta_strb pulse width	4	8	CPU cycles	
M31	ta_strb rising to chip select/address change	4	10	CPU cycles	
M32	data setup to ta_strb rising	0		ns	

Downloaded from www.Manualslib.com manuals search engine

Slow peripheral acknowledge read



Slow peripheral acknowledge write



Downloaded from www.Manualslib.com manuals search engine

Ethernet timing

All AC characteristics are measured with 10pF, unless otherwise noted.

The table below describes the values shown in the Ethernet timing diagrams.

Parm	Description	Min	Max	Unit	Notes
E1	MII tx_clk to txd, tx_en, tx_er	3	11	ns	2
E2	MII rxd, rx_en, rx_er setup to rx_clk rising	3		ns	
E3	MII rxd, rx_en, rx_er hold from rx_clk rising	1		ns	
E4	mdio (input) setup to mdc rising	10		ns	
E5	mdio (input) hold from mdc rising	0		ns	2
E6	mdc to mdio (output)	18	34	ns	1, 2
E7	mdc period	80		ns	3

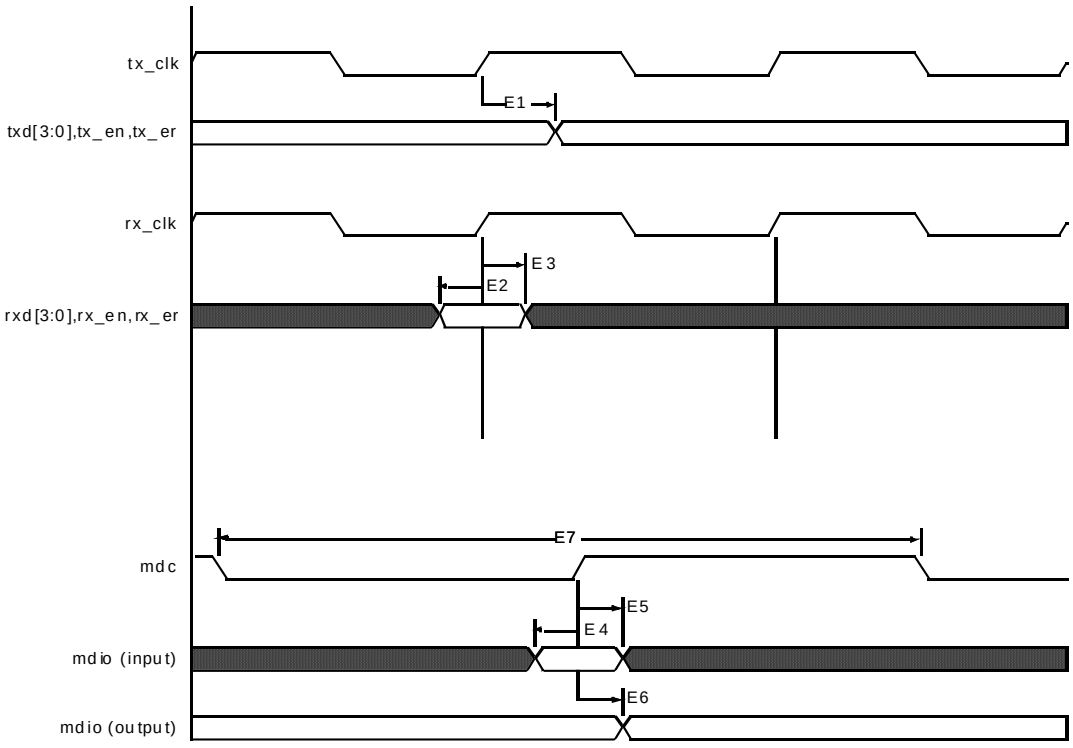
Notes:

1 Minimum specification is for fastest AHB bus clock of 88.5 MHz. Maximum specification is for slowest AHB bus clock of 51.6 MHz.

2 C_{load} = 10pf for all outputs and bidirects.

3 Minimum specification is for fastest AHB clock at 88.5 MHz.

Ethernet MII timing



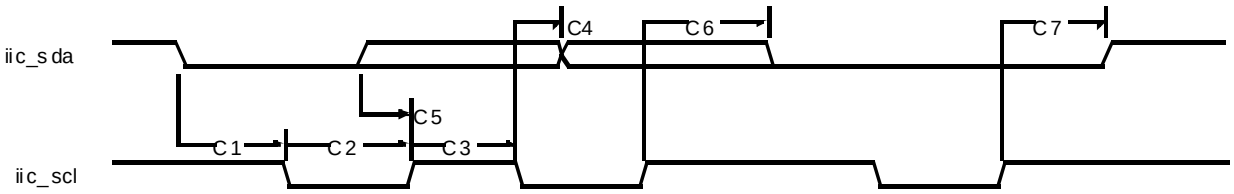
Downloaded from www.Manualslib.com manuals search engine

I²C timing

All AC characteristics are measured with 10pF, unless otherwise noted.

The table below describes the values shown in the I²C timing diagram.

		Standard Mode		Fast Mode		
Parm	**Description**	**Min**	**Max**	**Min**	**Max**	**Unit**
C1	iic_sda to iic_scl START hold time	4.0		0.6		µ
C2	iic_scl low period	4.7		1.3		µ
C3	iic_scl high period	4.7		1.3		µ
C4	iic_scl to iic_sda DATA hold time	0		0		µ
C5	iic_sda to iic_scl DATA setup tim	250		100		µ
C6	iic_scl to iic_sda STA	4.7		0.6		µ
C7	iic_scl to iic_sda STOP setup time	4.0		0.6		µ



Downloaded from www.Manualslib.com manuals search engine

SPI Timing

All AC characteristics are measured with 10pF, unless otherwise noted.

The next table describes the values shown in the LCD timing diagrams.

Parm	Description	Min	Max	Unit	Modes	Notes
SPI master parameters						
SPO	SPI enable low setup to first SPI CLK out rising	$3*T_{BCLK}$-10		ns	0,3	1,3
SP1	SPI enable low setup to first SPI CLK out falling	$3*T_{BCLK}$-10		ns	1,2	1,3
SP3	SPI data in setup to SPI CLK out rising	30		ns	0,3	
SP4	SPI data in hold from SPI CLK out rising	0		ns	0,3	
SP5	SPI data in setup to SPI CLK out falling	30		ns	1,2	
SP6	SPI data in hold from SPI CLK out falling	0		ns	1,2	
SP7	SPI CLK out falling to SPI data out valid		10	ns	0,3	6
SP8	SPI CLK out rising to SPI data out valid		10	ns	1,2	6
SP9	SPI enable low hold from last SPI CLK out falling	$3*T_{BCLK}$-10	+2	ns	0,3	1,3
SP1O	SPI enable low hold from last SPI CLK out rising	$3*T_{BCLK}$-10		ns	1,2	1,3
SP11	SPI CLK out high time	SP13*45%	SPI3*55%	ns	0,1,2,3	4
SP12	SPI CLK out low time	-SP13*45%	SPI3*55%	ns	0,1,2,3	4
SP13	SPI CLK out period	T_{BCLK}*6		ns	0,1,2,3	3
SPI slave parameters						
SP14	SPI enable low setup to first SPI CLK in rising	30		ns	0,3	1
SP15	SPI enable low setup to first SPI CLK in falling	30		ns	1,2	1
SP16	SPI data in setup to SPI CLK in rising	0		ns	0,3	
SP17	SPI data in hold from SPI CLK in rising	60		ns	0,3	
SP18	SPI data in setup to SPI CLK in falling	0		ns	1,2	
SP19	SPI data in hold from SPI CLK in falling	60		ns	1,2	
SP20	SPI CLK in falling to SPI data out valid	20	70	ns	0,3	6
SP21	SPI CLK in rising to SPI data out valid	20	70	ns	1,2	6

Downloaded from www.Manualslib.com manuals search engine

Parm	Description	Min	Max	Unit	Modes	Notes
SP22	SPI enable low hold from last SPI CLK in falling	15		ns	0,3	1
SP23	SPI enable low hold from last SPI CLK in rising	15		ns	1,2	1
SP24	SPI CLK in high time	SP26*40%	SP26*60%	ns	0,1,2,3	5
SP25	SPI CLK in low time	SP26*40%	SP26*60%	ns	0,1,2,3	5
SP26	SPI CLK in period	T_{BCLK}*8		ns	0,1,2,3	

Notes:

1 Active level of SPI enable is inverted (that is, 1) if the CSPOL bit in Serial Channel Control Register B is set to a 1. Note that in SPI slave mode, only a value of 0 (low enable) is valid; the SPI slave is fixed to an active low chip select.

2 SPI data order is reversed (that is, LSB last and MSB first) if the BITORDR bit in Serial Channel Control Register B is set to a 0.

3 T_{BCLK} is period of AHB clock.

4 ± 5% duty cycle skew.

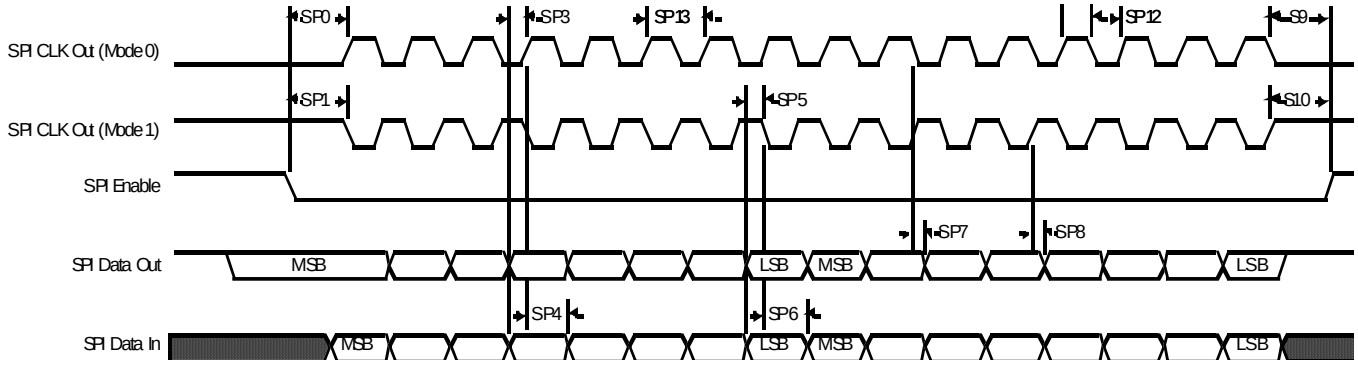
5 ± 10% duty cycle skew.

6 C_{load} = 5pf for all outputs.

7 SPI data order can be reversed such that LSB is first. Use the BITORDR bit in Serial Channel B/A/C/D Control Register A.

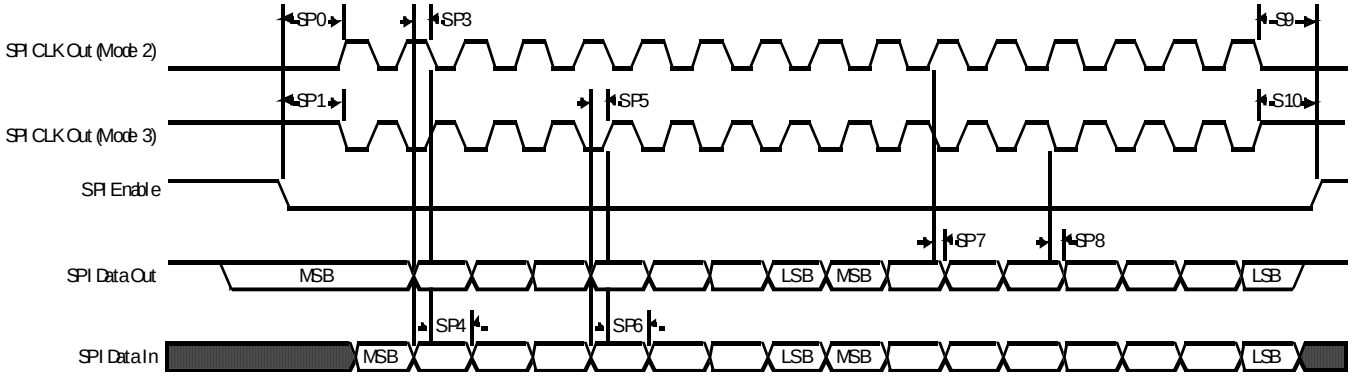
Downloaded from www.Manualslib.com manuals search engine

SPI master mode 0 and 1: 2-byte transfer



Note: SPI data can be reversed such that LSB is first. Use the BITORDER bit in Serial Channel B/A/C/D Control Register A.

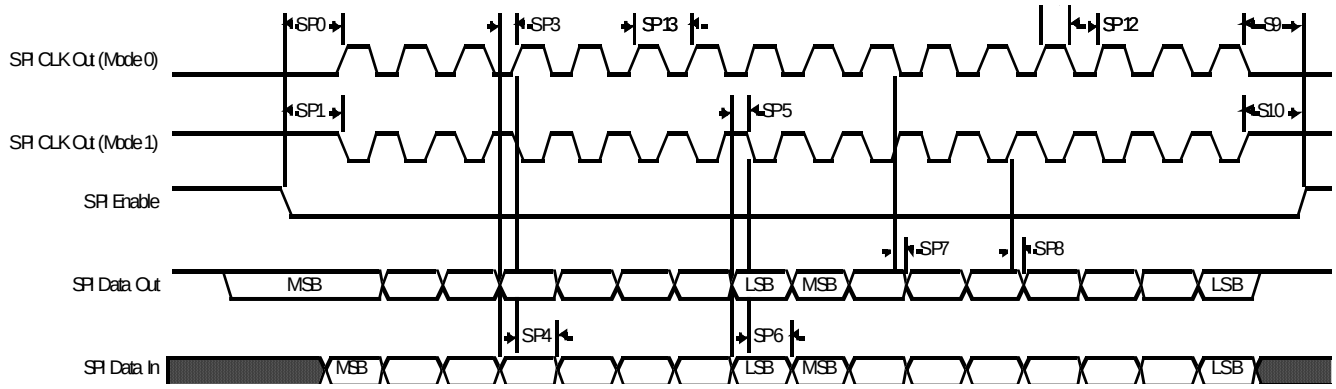
SPI master mode2 and 3: 2-byte transfer



Note: SPI data can be reversed such that LSB is first. Use the BITORDER bit in Serial Channel B/A/C/D Control Register A.

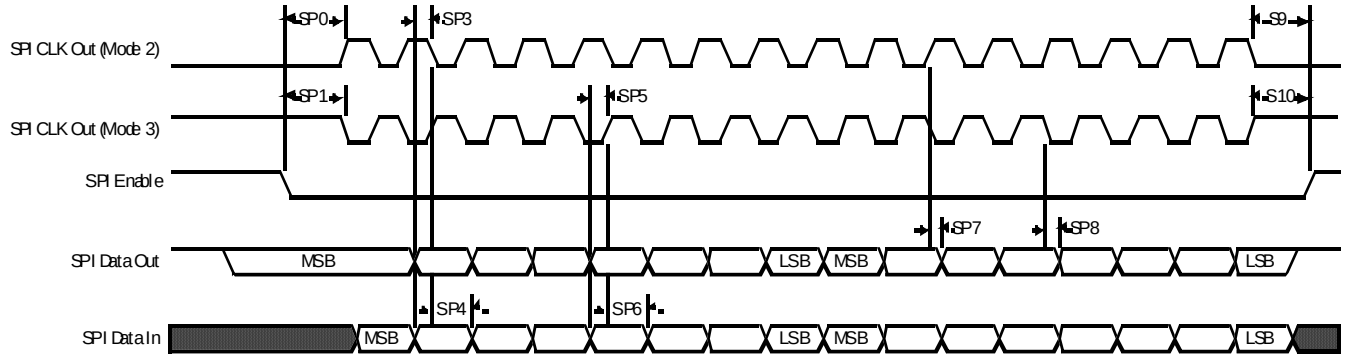
Downloaded from www.Manualslib.com manuals search engine

SPI slave mode 0 and 1: 2-byte transfer



Note: SPI data can be reversed such that LSB is first. Use the BITORDER bit in Serial Channel B/A/C/D Control Register A.

SPI slave mode 2 and 3: 2-byte transfer



Note: SPI data can be reversed such that LSB is first. Use the BITORDER bit in Serial Channel B/A/C/D Control Register A.

Downloaded from www.Manualslib.com manuals search engine

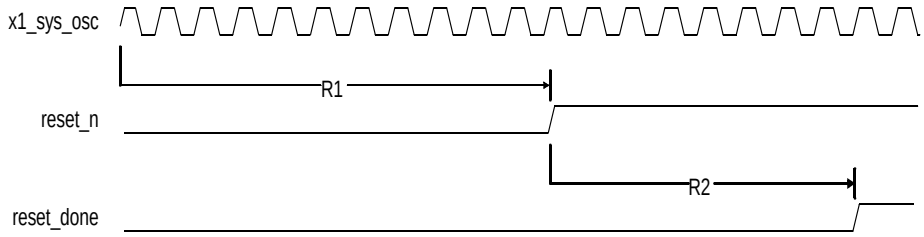
Reset and hardware strapping timing

All AC characteristics are measured with 10pF, unless otherwise noted.

The next table describes the values shown in the IEEE 1284 timing diagram.

Parm	Description	Min	Typ	Unit	Notes
R1	reset_n minimum time	10		x1_sys_osc clock cycles	1
R2	reset_n to reset_done		NOR flash: 4.5 SPI flash: 15	ms	

Note: The hardware strapping pins are latched 5 clock cycles after reset_n is deasserted (goes high).



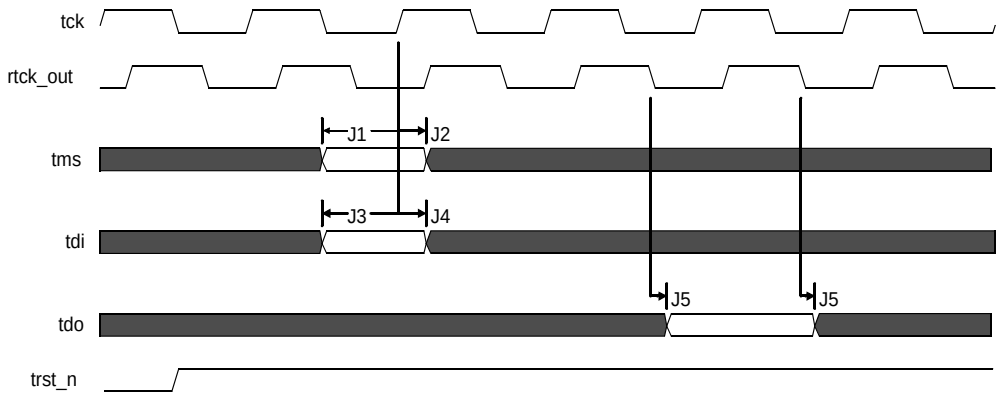
Downloaded from www.Manualslib.com manuals search engine

JTAG timing

All AC characteristics are measured with 10pF, unless otherwise noted.

The next table describes the values shown in the JTAG timing diagram.

Parm	Description	Min	Max	Unit
J1	tms (input) setup to tck rising	5		ns
J2	tms (input) hold to tck rising	2		ns
J3	tdi (input) setup to tck rising	5		ns
J4	tdi (input) hold to tck rising	2		ns
J5	tdo (output) to tck falling	2.5	10	ns



Notes:

1 Maximum tck rate is 10 MHz.

2 rtck_out is an asynchronous output, driven off of the CPU clock.

3 trst_n is an asynchronous input.

Downloaded from www.Manualslib.com manuals search engine

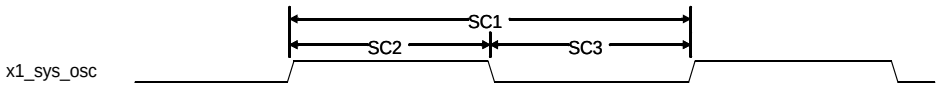
Clock timing

All AC characteristics are measured with 10pF, unless otherwise noted.

System PLL reference clock timing

Parm	Description	Min	Max	Unit	Notes
SC1	x1_sys_osc cycle time	25	50	ns	
SC2	x1_sys_osc high time	(SC1/2) x 0.45	(SC1/2) x 0.55	ns	
SC3	x1_sys_osc low time	(SC1/2) x 0.45	(SC1/2) x 0.55	ns	

The diagram below pertains to clock timing.



Downloaded from www.Manualslib.com manuals search engine

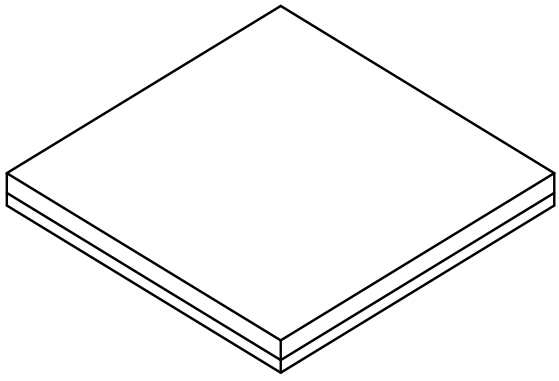
Downloaded from www.Manualslib.com manuals search engine

Packaging

CHAPTER 17

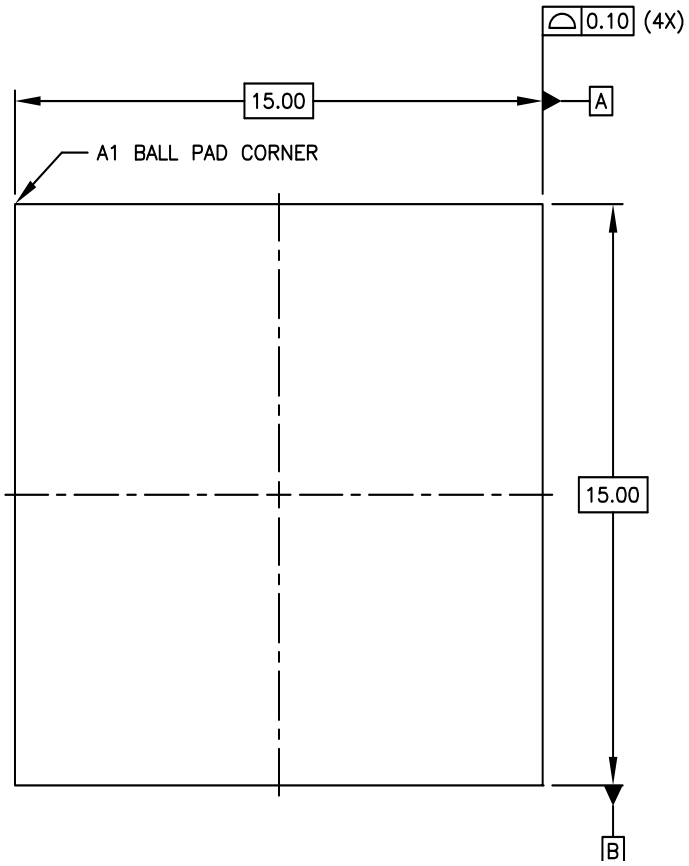
Below is the processor package, 265 LF-XBGA. Diagrams that follow show the processor dimensions: top, bottom, and side views.

Package

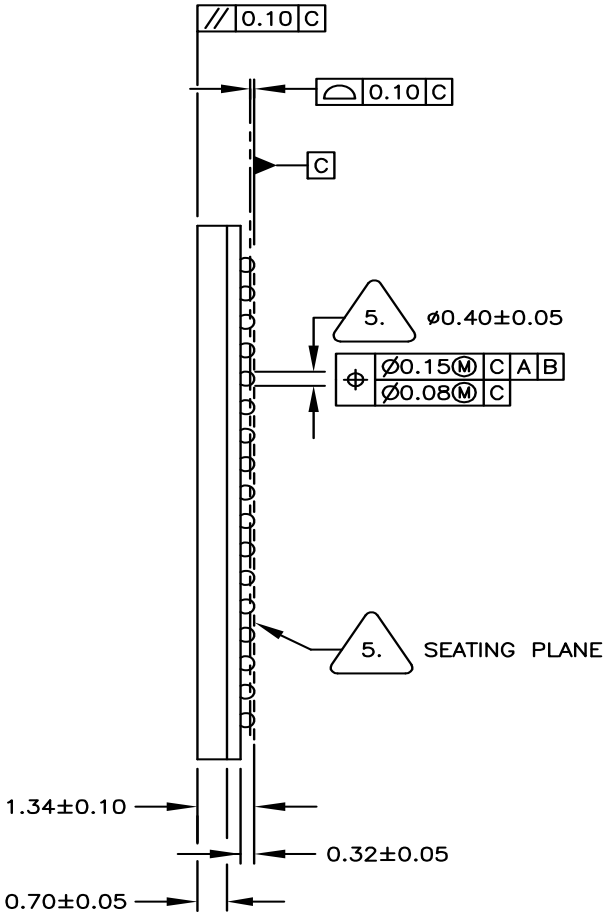


Downloaded from www.Manualslib.com manuals search engine

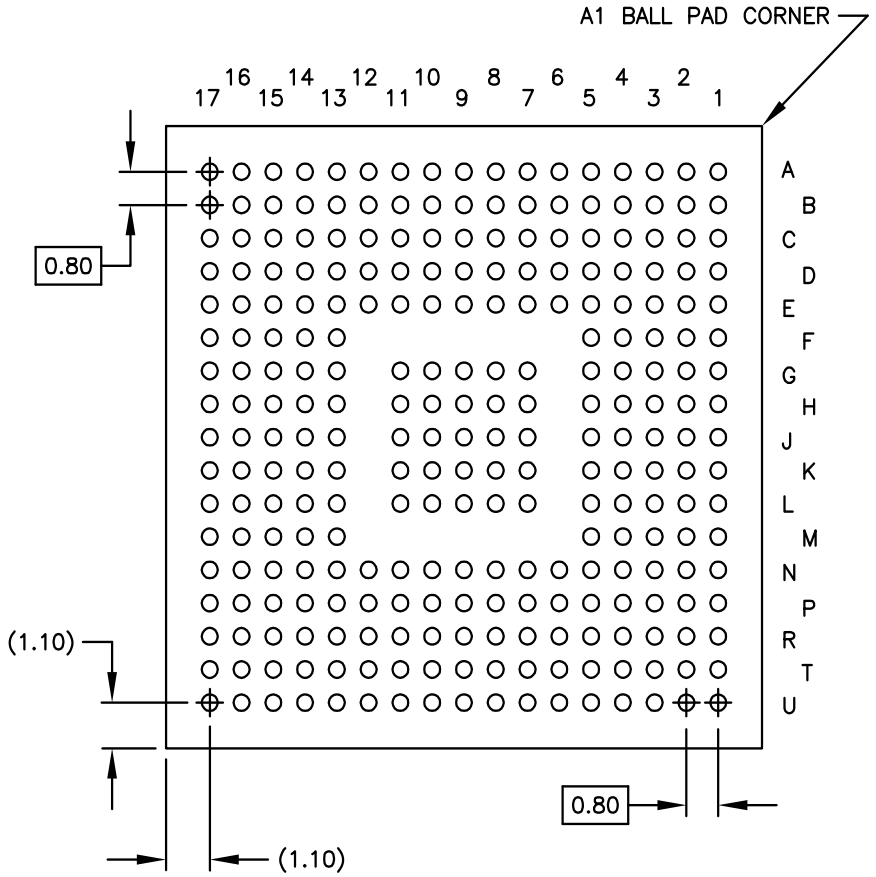
Processor Dimensions



Downloaded from www.Manualslib.com manuals search engine



Downloaded from www.Manualslib.com manuals search engine



Downloaded from www.Manualslib.com manuals search engine

Change log

CHAPTER 18

The following changes were made since the last revision of this document.

Revision B

Modified ADC data in the POR table.

Added RTC clock and battery back up connection information.

Updated POR and battery backup logic information for situations when the POR feature is not used.

Added power dissipation data for 75MHz.

Deleted IDDS because it does not apply to this type of IC.

Revision C

Added Flexible Interface Module signals to include PIC signals within GPIO pin out signals table.

Downloaded from www.Manualslib.com manuals search engine